ABSTRACT

Title of dissertation: NUMERICAL SOLUTION OF EIGENVALUE PROBLEMS WITH SPECTRAL TRANSFORMATIONS

Fei Xue, Doctor of Philosophy, 2009

Dissertation directed by: Professor Howard C. Elman
Department of Computer Science
Institute for Advanced Computer Studies

This thesis is concerned with inexact eigenvalue algorithms for solving large and sparse algebraic eigenvalue problems with spectral transformations. In many applications, if people are interested in a small number of interior eigenvalues, a spectral transformation is usually employed to map these eigenvalues to dominant ones of the transformed problem so that they can be easily captured. At each step of the eigenvalue algorithm (outer iteration), the matrix-vector product involving the transformed linear operator requires the solution of a linear system of equations, which is generally done by preconditioned iterative linear solvers inexactly if the matrices are very large. In this thesis, we study several efficient strategies to reduce the computational cost of preconditioned iterative solution (inner iteration) of the linear systems that arise when inexact Rayleigh quotient iteration, subspace iteration and implicitly restarted Arnoldi methods are used to solve eigenvalue problems with spectral transformations. We provide new insights into a special type of preconditioner with "tuning" that has been studied in the literature and propose new approaches to use tuning for solving the linear systems in this context. We also investigate other strategies specific to eigenvalue algorithms to further reduce the inner iteration counts. Numerical experiments and analysis show that these techniques lead to significant savings in computational cost without affecting the convergence of outer iterations to the desired eigenpairs.

NUMERICAL SOLUTION OF EIGENVALUE PROBLEMS WITH SPECTRAL TRANSFORMATIONS

by

Fei Xue

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:

Professor Howard Elman, Chair/Advisor
Professor Radu Balan
Professor Elias Balaras
Professor David Levermore
Professor Dianne O'Leary
Professor James Baeder, Dean's Representative

Acknowledgments

I owe great debt to many people who helped me in all aspects of life during the past few years at Maryland. I would like to express sincere gratitude to all of those who made this thesis possible.

First and foremost, I am indebted to my advisor, Professor Howard Elman, for providing with me exceptional guidance, encouragement and support throughout my graduate study. He taught me the first course in scientific computing, aroused my interest in this area and gave me an invaluable opportunity to work on an attractive and challenging topic. Our discussion on eigenvalue computation and other branches of scientific computing is always enlightening and rewarding. He was very patient and dedicated to help me improve academic writing, and gave me much freedom to develop my own intuitions, viewpoints and interests. He is not only an extraordinary advisor, but also a nice and lovable friend. I have been deeply inspired by his beliefs and values, both in research and everyday life.

I am very grateful to Dr. Melina Freitag and her advisor, Professor Alastair Spence at the University of Bath, with whom I have essentially worked in close collaboration. Their groundbreaking contribution to the study of inexact eigenvalue algorithms is fundamental to my thesis. Many of the ideas I developed in this thesis originated from fruitful discussions with them. Their suggestions and encouragement considerably motivated my efforts. I also thank Professor Valeria Simoncini at Università di Bologna for her insightful pioneering work in this area, and her careful reading of the first part of my work.

I would also like to acknowledge help and support from some professors. Professor Dianne O'Leary taught me three courses in scientific computing and has always been informative and supportive. She made quite some helpful comments on my thesis that help improve the thesis considerably. Professor emeritus G. W. Stewart wrote a book on eigenvalue algorithms which is my mostly used reference. His knowledge and understanding of eigenvalue problems is an encyclopedia to my study of this topic. Professor Daniel Szyld at Temple University, Professor Misha Kilmer at Tufts University and associate professor Eric de Sturler at Virginia Tech also have provided very useful information to my thesis.

I owe special thanks to my wife, Dandan Zhao, who has always stood by my side. Her love, encouragement, enthusiasm and optimism gave me great happiness, strength and confidence through my pursuit of career. Almost everything I achieved today is due to her everlasting support in our daily life. Words can never express my gratitude to her. Thanks to my parents, who cultivated my interest and dedication in research. Though they are thousands of miles away, their unlimited love and care have been great motivation for my career endeavors.

Finally, I would also like to thank some people at the University of Maryland. Thanks to Yi Li for being my best roomate for three years. Two good friends of mine, Ning Jiang and Weigang Zhong, who graduated from the AMSC program a few years ago, were very helpful during my first two years at Maryland. The AMSC program coordinator Alverda McCoy, business manager Sharon Welton and payroll coordinator Jodie Gray gave me significant help in administrative issues through my graduate study.

Table of Contents

List of Tables

List of Figures

List of Algorithms

1 Introduction

The theory and computation of eigenvalue problems are among the most successful and widely used tools of applied mathematics and scientific computing. Matrix eigenvalue problems arise naturally from a wide variety of scientific and engineering applications, including acoustics, control theory, earthquake engineering, graph theory, Markov chains, pattern recognition, quantum mechanics, stability analysis and many other areas. For a partial list of these applications, see [69, 91]. The increasing number of applications and the ever-growing scale of the problems have motivated fundamental progress in the numerical solution of eigenvalue problems in the past few decades. New insights and extensions of existing computational methods usually go hand in hand with the development of new algorithms and software packages.

The current state of the art is that excellent numerical methods have been used with great success for decades for dense matrices of small to medium size. The QR and QZ algorithms for standard eigenvalue problems $Av = \lambda v$ and generalized problems $Av = \lambda Bv$ are available in MATLAB [50], LAPACK [1] and many other commercial and public software packages. These algorithms are designed to compute the *complete* set of eigenvalues with full accuracy, and for problems of order n, they require $O(n^3)$ floating point operations and storage of size $O(n^2)$. Excellent introductions to the two methods can be found in [32, 86].

However, the QR and QZ algorithms are not practical for large and sparse eigenvalue problems, where only a small number of eigenvalues and eigenvectors are usually desired. The main reason is that the time complexity and storage requirements become very prohibitive for large n. Savings of the computational cost cannot be achieved if only a few eigenpairs are needed, as these methods cannot be adapted to compute only a partial set of eigenvalues. In addition, these algorithms do not take advantage of matrix structure such as sparsity and they cannot be used in circumstances where the matrices are not formed and stored explicitly; in this case, the only operation that can be performed on them is the matrix-vector multiplication.

To compute the desired eigenpairs of large sparse matrices, people have designed and implemented a large variety of eigenvalue algorithms based on the techniques of subspaces and projections. The general framework of these methods is to generate a sequence of subspaces $\mathcal{V}_1, \mathcal{V}_2, ...$ of small dimensions commensurate in size with the number of desired eigenvalues and project the large matrices onto these small subspaces; see [71] for a unified description of this approach. Ideally, the subspaces generated are expected to contain increasingly better approximations to the desired eigenvectors, and therefore some eigenvalues of the small projected matrices become progressively more accurate approximations to the desired eigenvalues. A most commonly used projection method is the Rayleigh-Ritz procedure; see [3, 86] for details.

The subspace-based methods differ from each other in the ways to generate the sequence of subspaces. One class of algorithms, which includes both single-vector and multiple-vector iterations, work with subspaces of fixed dimensions. These classical eigenvalue algorithms include the power method, inverse power method, Rayleigh quotient iteration [15, 32, 62, 94] and subspace iteration (also called orthogonal or simultaneous iteration) [15, 32, 69, 86]. Starting from a subspace $\mathcal{V}_k$, these methods generate the next subspace $\mathcal{V}_{k+1}$ of the same dimension by applying a linear operator $\mathcal{A}$ on $\mathcal{V}_k$. As k increases, $\mathcal{V}_k$ contains better approximate eigenvectors corresponding to the eigenvalues of $\mathcal{A}$ with largest magnitude (referred to as *dominant* eigenvalues).

A second class of methods use subspaces whose dimensions increase as the iteration proceeds. These methods are in general more efficient than fixed-dimension methods. A most important subclass is the Krylov subspace methods [3, 32, 86]. Starting from a single vector space $\mathcal{K}_1(\mathcal{A}, v) = \text{span}\{v\}$, Krylov subspace methods expand the Krylov subspace $\mathcal{K}_k(\mathcal{A}, v) = \text{span}\{v, \mathcal{A}v, \mathcal{A}^2v, ..., \mathcal{A}^{k-1}v\}$ in the k-th iteration to $\mathcal{K}_{k+1}(\mathcal{A}, v)$;

the new member vector of $\mathcal{K}_{k+1}(\mathcal{A}, v)$ is generated by applying $\mathcal{A}$ to the last member of $\mathcal{K}_k(\mathcal{A}, v)$. It is known that as the subspace dimension k increases, $\mathcal{K}_k(\mathcal{A}, v)$ generally contains quickly improving approximations to eigenvectors corresponding to *extremal* eigenvalues of $\mathcal{A}$; these eigenvectors are sometimes called "extremal eigenvectors" for convenience. Here, extremal eigenvalues refer to those located near both ends of the spectrum of a Hermitian $\mathcal{A}$, or those near the boundary of the convex hull of the spectrum of a non-Hermitian $\mathcal{A}$; they are also called exterior or peripheral eigenvalues. Dominant eigenvalues belong to the set of extremal eigenvalues.

For Hermitian problems, the most important Krylov subspace eigenvalue algorithm is the Lanczos method [44, 60]. This method computes a set of orthonormal basis vectors of $\mathcal{K}_k(\mathcal{A}, v)$ through the well-known three-term recurrence relation [32]. The superior convergence rate of approximate extremal eigenvalues was studied by Kaniel [42] and Paige [60]. It was later shown by Saad [70] that as k increases, $\mathcal{K}_k(\mathcal{A}, v)$ contains approximations that converge at least linearly to the eigenvectors of $\mathcal{A}$ corresponding to extremal eigenvalues, though the actual convergence rate can be faster [86]. For non-Hermitian problems, the Arnoldi method [2] was first presented by Saad [72] as an effective eigenvalue algorithm. This method generates an orthonormal set of basis vectors of the Krylov subspace by the k-term recurrence formula [32], where the cost of orthogonalization increases as the method proceeds. A convergence result of eigenvector approximation of the Arnoldi method can be found in [72, 69]. In addition, some nonsymmetric Lanczos methods are also used occasionally for non-Hermitian problems. The advantage of these methods is their fixed cost per iteration: they generate two sequences of Krylov subspaces through the three-term recurrence of the biorthogonalization process. However, their convergence properties are not as well-developed as those of the Hermitian Lanczos and the Arnoldi method; see [37, 38] for a survey. Both the Arnoldi and the non-Hermitian Lanczos methods naturally reduce to the regular Lanczos method for Hermitian problems.

For Hermitian problems where eigenvectors are needed or non-Hermitian problems, as CPU time and memory needed to manage the Krylov subspace increase with its dimension, a subspace restarting strategy is necessary. Roughly speaking, the restarting strategy builds a new subspace of smaller dimension by extracting the desired approximate eigenvectors from the current subspace of larger dimension. For the Arnoldi method, an elegant implicit restarting strategy based on the shifted-QR algorithm [32, 86] was proposed by Sorensen [85]. This method generates a new Krylov subspace of smaller dimension without using matrix-vector products involving $\mathcal{A}$. The resulting implicitly restarted Arnoldi (IRA) method has been implemented in ARPACK [49], a software package of high quality that has become the standard solver for practical large-scale eigenvalue problems.

Another subclass of algorithms generates non-Krylov type subspaces of increasing dimensions, such as the Davidson method [12] and the Jacobi-Davidson method [3, 22, 83]. For example, at each iteration step, the Jacobi-Davidson method expands the subspace by adding the solution of a *correction equation*. This newly added vector is orthogonal to the current subspace and provides the current desired eigenvector with a correction direction that comes from an approximate Newton iteration. The eigenvalue residual is expected to be significantly reduced with the new corrected approximate eigenvector. The restarting strategy is more straightforward for the Jacobi-Davidson method than for the Arnoldi method since no special structure of the non-Krylov subspace needs to be preserved.

It is well known that both the single and multiple-vector iterations and the Krylov subspace methods provide good approximations quickly to well-separated extremal eigenvalues [3, 86]. In many cases, however, the desired eigenvalues are not well-separated, or they are located in the interior of the spectrum, or they are substantially smaller in magnitude than other eigenvalues. In these situations, a spectral transformation is usually employed to map these eigenvalues to well-separated extremal ones of a transformed problem. Two commonly used spectral transformations are the shift-invert ($\mathcal{A}_S = (A - \sigma B)^{-1}B$) and Cayley transformations ($\mathcal{A}_C = (A - \sigma_1 B)^{-1}(A - \sigma_2 B)$); see [53] for details. The main difficulty associated with this approach is that at each step of the eigenvalue algorithm (outer iteration), a matrix-vector product involving the transformed operator $\mathcal{A}_S$ or $\mathcal{A}_C$ requires the solution of a linear system of the form $(A - \sigma B)x = y$. For some large-scale applications, for example, finite element discretization of three-dimensional partial differential equations, the matrices are so large that direct linear solvers based on matrix factorizations (e.g., sparse LU or Cholesky) are too expensive to apply. In these situations, it is necessary to use iterative methods (inner iteration) to solve these linear systems to some prescribed tolerances. This is the motivation to use inexact eigenvalue algorithms with "inner-outer" structure. The main focus of this thesis is to analyze some existing techniques and study new approaches to improve the effectiveness of various inexact eigenvalue algorithms, such as Rayleigh quotient iteration, subspace iteration and the implicitly restarted Arnoldi method, when they are used to solve eigenvalue problems with spectral transformations.

Clearly, the effectiveness of inexact eigenvalue algorithms strongly depends on that of the inner iteration. The fundamental approach to enhance the inner iteration efficiency is to use a proper iterative linear solver with a strong preconditioner, so that the linear systems can be solved to prescribed tolerances in a small or moderate number of steps. The most popular class of general purpose iterative linear solvers is also based on Krylov subspaces, which produces an approximate solution from the Krylov subspaces constructed from the system coefficient matrix and the right-hand side. Typical Krylov subspace solvers include the Conjugate Gradient method (CG) for Hermitian positive-definite systems, the Minimum Residual method (MINRES) for Hermitian indefinite systems, and the Generalized Minimum Residual method (GMRES) and variant biorthogonalization methods for non-Hermitian systems. The efficiency and robustness of iterative linear solvers can be improved by preconditioning. For example, to solve the linear system $Gz = b$, a preconditioner P can be used to transform the linear system to $GP^{-1}\tilde{z} = b$ and $z = P^{-1}\tilde{z}$. In general, P should approximate G in some sense and be such that it is inexpensive to solve linear systems $Px = y$. With appropriate preconditioning, iterative solvers applied to the preconditioned linear system are expected to converge significantly faster. In fact, the efficiency and reliability of iterative techniques usually depend much more on the quality of the preconditioner than on the specific choice of a Krylov subspace method. In this thesis, we assume that a reasonably good preconditioner P is already available for solving the linear systems arising in inexact eigenvalue algorithms. For a comprehensive introduction to Krylov subspace linear solvers and preconditioning techniques, we refer to [36, 68].

In some situations, however, the performance of preconditioners for solving linear systems in general settings may not be indicative of their effectiveness in the setting of eigenvalue computation. The behavior of inner solves may be changed significantly by certain minor modifications of the preconditioner. Some other factors can also have significant effects on the performance of inner solves arising in inexact eigenvalue algorithms. For instance, the tolerance for the approximate solution of the linear system in some outer iterations may be relaxed without obviously affecting the convergence rate of the eigenvalue algorithm. These important issues associated with inexact eigenvalue algorithms must be carefully studied and properly handled so that the inner iteration counts can be adequately reduced.

For example, suppose inexact Rayleigh quotient iteration (RQI) is used to compute the lowest eigenpair of a Hermitian positive definite matrix. When solving the linear systems arising in inexact RQI, regular preconditioned MINRES does not perform as well as what is expected from its performance in the usual setting of solving Hermitian indefinite systems. To overcome this difficulty, a special low-rank modification (also called "tuning") of preconditioning operators was proposed in [26]. The function of tuning is to change the preconditioning operator slightly so that the right-hand side of the linear system is an approximate eigenvector of the preconditioned coefficient matrix. Though tuning does not reduce the effective condition number of the preconditioned operator, this modification leads to considerably improved performance of MINRES algorithm in this setting. The first major part of the thesis gives a detailed analysis of the performance of preconditioned MINRES for solving the linear systems arising in inexact RQI. We provide new perspectives on the difficulties of preconditioned MINRES without tuning and show how tuning improves the performance. We also explore an initial period of slow convergence exhibited by MINRES in this context. We show that if tuning is applied, this initial period of slow convergence becomes longer as the outer iteration proceeds, but the rate at which the MINRES iterate converges to the desired eigenvector does not slow down; it only depends on a certain "effective condition number" of the preconditioned system matrix.

To study inexact eigenvalue algorithms to compute a few eigenpairs of generalized eigenvalue problems, without loss of generality, we assume that the transformed operator can be expressed in the generic form $\mathcal{A} = A^{-1}B$. For inexact subspace iteration, a linear system with multiple right-hand sides (block system), namely $AY^{(i)} = BX^{(i)}$, needs to be solved in the ith outer iteration, where $X^{(i)}$ is the approximate desired invariant subspace therein. It is shown in [65] that to retain the linear convergence of the outer iteration, $AY^{(i)} = BX^{(i)}$ need be solved to only a modest accuracy in the first few outer iterations (when i is small), but the accuracy requirement (tolerance) needs to be made gradually more stringent as the outer iteration proceeds. As a result, the inner iteration counts keep increasing with the outer iteration. It is shown in [65] that tuning can also be applied to make the right-hand side $BX^{(i)}$ an approximate invariant subspace of the preconditioned coefficient matrix and consequently keep the inner iteration counts from increasing. In the second part of thesis, we propose a new two-phase strategy to solve $AY^{(i)} = BX^{(i)}$: in the first phase, an approximate solution $Y_1^{(i)}$ is obtained by applying a single step of block-GMRES iteration with tuning to $AY^{(i)} = BX^{(i)}$, or by solving an inexpensive least squares problem; in the second phase, a correction equation $A\,dY^{(i)} = BX^{(i)} - AY_1^{(i)}$ is solved by block linear solvers *without* tuning to a *fixed* relative tolerance, and $Y^{(i)} \leftarrow Y_1^{(i)} + dY^{(i)}$ is the solution to $AY^{(i)} = BX^{(i)}$. We show that this algorithm also keeps the inner iteration counts from increasing, and we discuss its close connection to the inverse correction scheme [33, 67] and the residual inverse power method [87]. In addition, we study a few additional enhancements that can be applied in the second phase to further decrease the inner iteration counts. One of the enhancements is the use of subspace recycling [61] with iterative linear solvers to efficiently solve a sequence of linear systems. Numerical experiments show that the inner iteration cost tends to decrease with the progress of outer iterations if these strategies are used all together.

The two-phase strategy can also be applied to the linear systems arising in an inexact implicitly restarted Arnoldi (IRA) method. Specifically, at the jth Arnoldi step in the ith cycle of IRA, $Ay = Bu_{j+1}^{(i)}$ needs to be solved, where $u_{j+1}^{(i)}$ is the last Arnoldi vector in that step. We first propose and study a new tuning strategy constructed using solution vectors of previously solved linear systems in both the current and previous IRA cycles. We show that a single step of GMRES iteration with this tuned preconditioner applied to $Ay = Bu_{j+1}^{(i)}$ gives a good approximate solution y_1 that is roughly a linear combination of those solution vectors. With this approximate solution, the correction equation $A\,z = Bu_{j+1}^{(i)} - Ay_1$ can be solved without tuning to a tolerance much larger than that required for the original system $Ay = Bu_{j+1}^{(i)}$. Therefore, the inner iteration counts needed solve the correction equation can be considerably smaller than those required to solve $Ay = Bu_{j+1}^{(i)}$ without tuning. In addition, it is shown in [74, 28] that the allowable tolerances of $Ay = Bu_{j+1}^{(i)}$ can be relaxed as the IRA method converges to the desired invariant subspace. Consequently, the inner iteration counts needed to solve the linear system will decrease as the IRA cycle proceeds. To estimate the allowable tolerances, [28] gives a sufficient condition for the tolerances which guarantees the desired approximate invariant subspace will not be contaminated by the errors introduced in the solution of $Ay = Bu_{j+1}^{(i)}$. Our analysis, on the other hand, discusses a necessary condition for the tolerances, the violation of which necessarily leads to contamination of the desired approximate invariant subspace by excessive errors of inner solves. From this observation, we derive a theoretically more accurate estimate of the allowable tolerances which slightly outperforms the estimate from [28]. The use of subspace recycling to solve the correction equations is also discussed. Numerical experiments show that the combined use of these strategies significantly reduces the inner iteration cost.

The thesis is organized as follows. In Chapter 2, we review the basic theory, tools and solvers needed to study inexact eigenvalue algorithms, and we discuss some related work in the literature. Chapter 3 investigates the convergence of preconditioned MINRES with and without tuning for solving the linear systems arising in inexact Rayleigh quotient iteration. Chapter 4 provides some new insights into tuning and studies the new two-phase strategy and some additional enhancements to solve the block linear systems in inexact subspace iteration. Chapter 5 explores a new way to construct tuning for the two-phase strategy and gives a refined analysis of allowable tolerances for solving the linear systems in an inexact implicitly restarted Arnoldi method. Finally, in Chapter 6, we summarize the thesis and suggest some areas for future research.

2 Background

This chapter gives a brief review of the basic definitions, tools and theories needed to study inexact eigenvalue problems. This background introduction includes eigenvalue problems, Krylov subspace projection methods, eigenvalue algorithms, spectral transformations, preconditioned Krylov subspace linear solvers and related work in literature.

2.1 Basic definitions and tools of eigenvalue problems

In this section, we briefly review some basic definitions, properties and theories of algebraic eigenvalue problems. Let A be a $n \times n$ square matrix, λ a scalar, and v a nonzero column vector of length n, such that

$$Av = \lambda v. \tag{2.1}$$

This equation is referred to as the standard eigenvalue problem. Here, λ is an *eigenvalue* of A, v is the corresponding right *eigenvector*, and (λ, v) is called an *eigenpair*. Similarly, a left eigenvector is defined by the equation $w^*A = \lambda w^*$, where w^* is the conjugate transpose of w. Unless otherwise stated, the term eigenvector refers to the right eigenvector. In addition, we assume that eigenvectors are normalized, i.e., the norm of any eigenvector equals one. Throughout the thesis, we use the ordinary Euclidean norm $\|x\| = (x,x)^{1/2}$ for a vector $x \in \mathbb{C}^n$ and the induced 2-norm $\|F\| = \sup_{\|x\|=1} \|Fx\|$ for a matrix $F \in \mathbb{C}^{n\times n}$. If $G \in \mathbb{C}^{n\times n}$ is Hermitian positive definite, the G-norm of an vector x is defined as $\|x\|_G = \sqrt{x^*Gx}$.

Eigenvalues of A are the roots of the characteristic polynomial $\mathbf{p}(\lambda) = \det(\lambda I - A)$. An eigenvalue is called a *simple* one if it is a simple root of $\mathbf{p}(\lambda)$ (with algebraic multiplicity one); otherwise it is a *multiple* eigenvalue. The full set of eigenvalues of A is called the *spectrum* and is denoted by $\lambda(A) = \{\lambda_1, \lambda_2, ..., \lambda_n\}$. The spectrum of A remains invariant under similarity transformations, i.e., if X is square and nonsingular, then for any $\tilde{A} = X^{-1}AX$, $\lambda(A) = \lambda(\tilde{A})$.

A subspace $\mathcal{V}$ that satisfies $v \in \mathcal{V} \Rightarrow Av \in \mathcal{V}$ is called an *invariant subspace* (eigenspace) of A. An eigenvector spans a one-dimensional invariant subspace. A desired invariant subspace refers to a space spanned by the eigenvectors corresponding to a group of wanted eigenvalues.

If A has n linearly independent eigenvectors, it can be diagonalized as

$$A = V\Lambda V^{-1}, \tag{2.2}$$

where $V = [v_1, v_2, ..., v_n]$ contains the eigenvectors of A, and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$ is a diagonal matrix containing the corresponding eigenvalues. In particular, if A has distinct eigenvalues, it is diagonalizable. A Hermitian matrix A ($A^* = A$) is diagonalizable; it has only real eigenvalues and a complete orthonormal set of eigenvectors. The diagonalization is also called the *spectral decomposition.*

In addition to the spectral transformation, for any matrix A, there exists a *Schur decomposition* $A = UTU^*$ where U is a unitary matrix ($U^* = U^{-1}$) and the Schur form T is upper triangular. The diagonal entries of T are eigenvalues of A. The columns of U are called *Schur vectors.* By proper choice of U, the eigenvalues of A can appear in any order on the diagonal of T.

Let $A = UTU^*$ be the Schur decomposition of A, where $U = [U_1, U_2]$ with $U_1 \in \mathbb{C}^{n\times p}$, $T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}$ with $T_{11} \in \mathbb{C}^{p\times p}$ and $T_{22} \in \mathbb{C}^{(n-p)\times(n-p)}$. Then $AU_1 = U_1T_{11}$ is called a partial Schur decomposition of A. Here U_1 contains orthonormal columns that span the invariant subspace of A corresponding to the eigenvalues that appear on the diagonal of T_{11}.

A most important tool connecting a block triangular matrix with a block diagonal matrix is the Sylvester equation. For example, consider the block diagonalization of the matrix $T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}$. Suppose Q is the solution of the Sylvester equation $T_{11}Q - QT_{22} = -T_{12}$. Then

$$\begin{bmatrix} I_p & -Q \\ 0 & I_{n-p} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} I_p & Q \\ 0 & I_{n-p} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{11}Q - QT_{22} + T_{12} \\ 0 & T_{22} \end{bmatrix} = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix}. \tag{2.3}$$

In general, given $K \in \mathbb{C}^{p\times p}$ and $M \in \mathbb{C}^{q\times q}$, the Sylvester operator $\mathcal{S} : \mathbb{C}^{p\times q} \to \mathbb{C}^{p\times q}$ associated with these two matrices is defined as a linear transformation $\mathcal{S} : G \to \mathcal{S}(G) = KG - GM$. This transformation is nonsingular if and only if $\lambda(K) \cap \lambda(M) = \emptyset$. The separation between K and M is defined as

$$\operatorname{sep}(K, M) = \inf_{\|G\|=1} \|KG - GM\|, \tag{2.4}$$

and the norm of $\mathcal{S}$ is defined as

$$\|\mathcal{S}\| = \sup_{\|G\|=1} \|KG - GM\|. \tag{2.5}$$

We have so far reviewed some preliminary definitions and tools to study standard eigenvalue problems. The definitions and tools of generalized eigenvalue problems $Av = \lambda Bv$, though more complicated, are largely parallel to what is presented in this section. In particular, the generalized problem is equivalent to the standard problem $B^{-1}Av = \lambda v$ for nonsingular B. To simplify the analysis, we assume throughout the thesis that B is nonsingular, unless otherwise stated.

2.2 The framework of Krylov subspace projection methods

2.2.1 Definition and basic properties of Krylov subspaces

Krylov subspaces are among the most widely used building blocks of iterative linear solvers and eigenvalue algorithms for large sparse matrices. Given a linear operator $\mathcal{A}$ and a nonzero initial vector u_1, the k-th order Krylov subspace is $\mathcal{K}_k(\mathcal{A}, u_1) = \text{span}\{u_1, \mathcal{A}u_1, ..., \mathcal{A}^{k-1}u_1\}$. The generation of Krylov subspaces only needs the operation of matrix-vector product involving $\mathcal{A}$. If $\mathcal{A}$ is a large sparse matrix with nnz nonzero entries, each matrix-vector product generating a new member vector of the Krylov subspace can be computed in only nnz floating point operations.

It can be readily shown that $\mathcal{K}_k(\mathcal{A}, u_1) \subset \mathcal{K}_{k+1}(\mathcal{A}, u_1)$, $\mathcal{A}\mathcal{K}_k(\mathcal{A}, u_1) \subset \mathcal{K}_{k+1}(\mathcal{A}, u_1)$, $\mathcal{K}_k(\mathcal{A}, u_1) = \mathcal{K}_k(s\mathcal{A}, u_1) = \mathcal{K}_k(\mathcal{A}, su_0)$ for any nonzero scalar s, and $\mathcal{K}_k(\mathcal{A}, u_1) = \mathcal{K}_k(\mathcal{A} - \sigma I, u_1)$ for any scalar σ.

An important property of Krylov subspaces is that the $\mathcal{K}_k(\mathcal{A}, u_1)$ contains quickly improving approximations to eigenvectors corresponding to extremal eigenvalues of $\mathcal{A}$. To simplify the introduction, we only present the result for Hermitian $\mathcal{A}$ with eigenvalues $\lambda_1 > \lambda_2 \geq ... \geq \lambda_n$. For a given eigenvector v_i of $\mathcal{A}$, the quality of the best approximation to v_i contained in $\mathcal{K}_k(\mathcal{A}, u_1)$ can be measured by the tangent of the angle between v_i and its orthogonal projection onto $\mathcal{K}_k(\mathcal{A}, u_1)$, denoted by $\tan \angle(v_i, \mathcal{K}_k(\mathcal{A}, u_1))$. It is obvious from the structure of the Krylov subspaces that any vector $v \in \mathcal{K}_k(\mathcal{A}, u_1)$ can be written as $\mathbf{q}_{k-1}(\mathcal{A})u_1$, where $\mathbf{q}_{k-1}$ is some polynomial of degree k–1. The problem of finding the orthogonal projection of v_i onto $\mathcal{K}_k(\mathcal{A}, u_1)$ is equivalent to finding a polynomial $\mathbf{q}_{k-1}$ for which $\tan \angle(v_i, \mathbf{q}_{k-1}(\mathcal{A})u_1)$ is minimized. This polynomial itself does not have a simple analytic form, but it is well known that the Chebyshev polynomial can be used to provide an upper bound of this angle. In fact, it is shown in [86] that

$$\tan\angle(v_1, \mathcal{K}_k(\mathcal{A}, u_1)) \leq \frac{\tan\angle(v_1, u_1)}{c_{k-1}(1+2\eta)} \lesssim \frac{\tan\angle(v_1, u_1)}{(1+2\sqrt{\eta+\eta^2})^{k-1}}, \tag{2.6}$$

where $\eta = \frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n}$, and $c_{k-1}(t) = (1+\sqrt{t^2-1})^{k-1} + (1+\sqrt{t^2-1})^{1-k}$ for $|t| > 1$ is the $(k-1)$-th order Chebyshev polynomial of the first kind. Similar results can be obtained for the approximation to v_2 and other eigenvectors. This bound shows that the best approximation contained in $\mathcal{K}_k(\mathcal{A}, u_1)$ to eigenvectors corresponding to extremal eigenvalues converges at least linearly. In practice, this convergence rate of eigenvector approximation tends to be superlinear, as observed in [86, 45]. The superlinear convergence in the setting of Krylov subspace linear solvers is studied in [76].

2.2.2 The Arnoldi and Lanczos processes

The original form of the Krylov subspace basis $\{u_1, \mathcal{A}u_1, ..., \mathcal{A}^{k-1}u_1\}$ becomes progressively ill-conditioned as k increases, because $\mathcal{A}^{k-1}u_1$ converges to the dominant eigenvector(s) of $\mathcal{A}$. To resolve this difficulty, the Arnoldi process [2] computes a set of orthonormal basis vectors for $\mathcal{K}_k(\mathcal{A}, u_1)$ as described in Algorithm 2.1.

Algorithm 2.1 The Arnoldi Process

```
Given a unit vector $u_1$, $U_k = [u_1]$, $\hat{H}_{k-1}$ is an empty $1 \times 0$ matrix
for $k = 1, 2, ...,$ do
    1. $h_k = U_k^* \mathcal{A} u_k$, $v = \mathcal{A}u_k - U_k h_k$
    2. $h_{k+1,k} = \|v\|_2$, $u_{k+1} = v/h_{k+1,k}$, $U_{k+1} = [U_k, u_{k+1}]$, $\hat{H}_k = \begin{bmatrix} \hat{H}_{k-1} & h_k \\ 0_{1\times(k-1)} & h_{k+1,k} \end{bmatrix}$
end for
```

In short, the Arnoldi process computes $\mathcal{A}u_k$, orthogonalizes it against U_k, and normalizes the result to u_{k+1}. This process gives the Arnoldi decomposition $\mathcal{A}U_k = U_{k+1}\hat{H}_k = U_k H_k + h_{k+1,k} u_{k+1} e_k^T$, where $H_k = U_k^* \mathcal{A} U_k \in \mathbb{C}^{k\times k}$ is an upper Hessenberg matrix containing the leading k rows of $\hat{H}_k = U_{k+1}^* \mathcal{A} U_k \in \mathbb{C}^{(k+1)\times k}$, and $e_k^T = (0, 0, ..., 0, 1) \in \mathbb{R}^k$. It can be shown readily that the orthonormal column vectors of U_k span $\mathcal{K}_k(\mathcal{A}, u_1)$.

For Hermitian $\mathcal{A}$, the Arnoldi process naturally reduces to the Lanczos process [44], where $\mathcal{A}u_k$ is automatically orthogonal to $u_1, ..., u_{k-2}$ and thus only needs to be orthogonalized against u_{k-1} and u_k and then normalized to u_{k+1}. This procedure is described by the well-known three term recurrence formula. Therefore the Lanczos decomposition is usually written as $\mathcal{A}U_k = U_kT_k + \beta_k u_{k+1}e_k^T$, where T_k is a tridiagonal real symmetric matrix with $\alpha_j = u_j^*\mathcal{A}u_j$ $(1 \leq j \leq k)$ on the main diagonal and $\beta_j = \|\mathcal{A}u_j - \alpha_j u_j - \beta_{j-1}u_{j-1}\|_2$ $(1 \leq j \leq k-1)$ on the sub- and super-diagonals. In floating point arithmetic, the loss of orthogonality of Lanczos vectors may prevent Ritz vectors from approximating the desired eigenvectors, and some reorthogonalization procedure is necessary to resolve this difficulty; see [86] and references therein.

2.2.3 Some projection methods

Given a Krylov subspace $\mathcal{K}_k(\mathcal{A}, u_1)$, an approximate eigenpair (μ, w) of $\mathcal{A}$ can be obtained by several projection methods which extract w from $\mathcal{K}_k(\mathcal{A}, u_1)$. The most commonly used method is the Rayleigh-Ritz procedure based on the Galerkin condition. Specifically, this condition requires that the eigenvalue residual vector be orthogonal to the Krylov subspace:

$$(2.7)\quad \mathcal{A}w - \mu w \perp \mathcal{K}_k(\mathcal{A}, u_1), \quad \text{or, equivalently} \quad v^*(\mathcal{A}w - \mu w) = 0 \;\; \forall v \in \mathcal{K}_k(\mathcal{A}, u_1).$$

Let $w = U_k y \in \mathcal{K}_k(\mathcal{A}, u_1)$, where the columns of U_k are orthonormal basis vectors of $\mathcal{K}_k(\mathcal{A}, u_1)$. The Galerkin condition can be translated into the matrix problem

$$U_k^*\mathcal{A}U_k y = \mu y. \tag{2.8}$$

The above derivation leads to the Rayleigh-Ritz procedure as given in Algorithm 2.2.

Algorithm 2.2 Rayleigh-Ritz procedure

```
1. Compute an orthonormal set of basis $U_k = [u_1, u_2, ..., u_k]$ of $\mathcal{K}_k(\mathcal{A}, u_1)$
2. Compute $H_k = U_k^* \mathcal{A} U_k$, i.e., the projection of $\mathcal{A}$ onto $\mathcal{K}_k(\mathcal{A}, u_1)$
3. Compute an eigenpair $(\mu, y)$ of $H_k$
4. $(\mu, U_k y)$ is an approximate eigenpair of $\mathcal{A}$
```

Here, μ is called a Ritz value, and $U_k y$ is the associated Ritz vector. The Rayleigh-Ritz procedure essentially projects the original large problem onto the small subspace $\mathcal{K}_k(\mathcal{A}, u_1)$ to get a small eigenvalue problem of order k. It naturally fits into the Arnoldi and Lanczos process, where U_k and H_k (or T_k) are computed.

Ritz pairs tend to approximate extremal eigenpairs better than interior ones. An analysis of the quality of the Ritz pair $(\mu, U_k y)$ as an approximate eigenpair of $\mathcal{A}$ can be found in [86].

The Rayleigh-Ritz procedure is based on the orthogonal projection where the eigenvalue residual $\mathcal{A}w - \mu w \perp \mathcal{K}_k(\mathcal{A}, u_1)$ and $w \in \mathcal{K}_k(\mathcal{A}, u_1)$. An approximate eigenpair $(\tilde{\mu}, \tilde{w})$ can also be obtained by an *oblique projection* where, for example, a Petrov-Galerkin condition $\mathcal{A}\tilde{w} - \tilde{\mu}\tilde{w} \perp \mathcal{A}\mathcal{K}_k(\mathcal{A}, u_1)$ with $\tilde{w} \in \mathcal{K}_k(\mathcal{A}, u_1)$ is imposed. The matrix form of this condition is

$$U_k^* \mathcal{A}^* \mathcal{A} U_k \tilde{y} = \tilde{\mu} U_k^* \mathcal{A}^* U_k \tilde{y}. \tag{2.9}$$

In the Arnoldi process, this equation can be written as $\tilde{H}_k^* \tilde{H}_k \tilde{y} = \tilde{\mu} H_k \tilde{y}$. The quantities $\tilde{\mu}$ and $U_k \tilde{y}$ are referred to as harmonic Ritz value and harmonic Ritz vectors. The formal introduction to harmonic Ritz pairs $(\tilde{\mu}, U_k \tilde{y})$ and their connection with Ritz pairs $(\mu, U_k y)$ is given in [63]. In general, harmonic Ritz pairs are preferred as approximations to interior eigenpairs; see Chapter 3 of [3].

The Ritz and harmonic Ritz pairs can also be defined and extracted from generic non-Krylov subspaces, for example, the subspaces generated by the Jacobi-Davidson method, in a similar way as they are constructed from the Krylov subspaces; see [3] for details.

2.3 Eigenvalue algorithms

In Section 2.2, we have outlined the framework of the Arnoldi and Lanczos methods for computing approximate eigenpairs of non-Hermitian and Hermitian matrices. These methods are based on a combination of the Arnoldi (or Lanczos) process that generates Krylov subspaces for the candidate approximate eigenvectors, and the Rayleigh-Ritz or Petrov-Galerkin procedure, which extracts the approximate eigenvectors from the subspaces. For a complete treatment of theoretical and computational background of important eigenvalue algorithms for large sparse matrices, especially the Jacobi-Davidson method, we refer to [3]. In this section, we briefly review three algorithms studied in detail in this thesis: Rayleigh quotient iteration, subspace iteration, and the implicitly restarted Arnoldi (IRA) method.

Rayleigh quotient iteration is a simple single-vector iteration for computing a simple eigenpair of a symmetric matrix A. For a given vector u, elementary calculus shows that the Rayleigh quotient $\sigma(u) = \frac{u^*Au}{u^*u}$ minimizes the eigenvalue residual norm $\|Au - \sigma(u)u\|$. Therefore, if u is an approximate eigenvector, then $\sigma(u)$ is a good corresponding approximate eigenvalue. Combining the Rayleigh quotient with the inverse power method leads to the Rayleigh quotient iteration.

Algorithm 2.3 Rayleigh quotient iteration (RQI)

```
Given $x^{(0)}$ with $\|x^{(0)}\| = 1$
for $i = 0, 1, ...,$ until convergence do
  1. Compute the Rayleigh quotient $\sigma^{(i)} \leftarrow \frac{x^{(i)*}Ax^{(i)}}{x^{(i)*}x^{(i)}}$
  2. Solve $(A - \sigma^{(i)}I)y^{(i)} = x^{(i)}$
  3. $x^{(i+1)} \leftarrow y^{(i)}/\|y^{(i)}\|$ and test for convergence
end for
```

It is shown in [62] that if $x^{(0)}$ is already a good approximate eigenvector, the rate of converge of RQI is cubic. Details about the practical implementation of this method can also be found in [62].

The Rayleigh quotient iteration described in Algorithm 2.3 can also be applied to compute a simple eigenpair of a non-Hermitian matrix A; the resulting method is usually called inverse iteration with Rayleigh quotient shift. In general, it is not applicable for computing a few eigenvalues. In this situation, subspace iteration (also called orthogonal or simultaneous iteration), a straightforward block generalization of the power method, can be used. A basic version of this algorithm is outlined in Algorithm 2.4.

Algorithm 2.4 Subspace iteration (basic version)

```
Given $X^{(0)} \in \mathbb{C}^{n \times p}$ with $X^{(0)*}X^{(0)} = I$
for $i = 0, 1, ...,$ until convergence do
  1. Compute $Y^{(i)} \leftarrow \mathcal{A}X^{(i)}$
  2. Orthogonalize $Y^{(i)}$ into $X^{(i+1)}$ and test for convergence
end for
3. Solve the projected small eigenvalue problem $X^{(i)*}AX^{(i)} = \mu w$;
the approximate eigenpairs of $A$ are $(\mu_j, X^{(i)}w_j)$
```

Subspace iteration computes simultaneously several dominant eigenpairs of the linear operator $\mathcal{A}$, whose eigenvalues satisfy $|\lambda_1| \leq ... \leq |\lambda_{n-p}| < |\lambda_{n-p+1}| \leq ... \leq |\lambda_n|$. It can be shown that span$\{X^{(i)}\}$ converges to the invariant subspace corresponding to $\{\lambda_1, \lambda_2, ..., \lambda_p\}$ at a linear rate $\left|\frac{\lambda_{n-p}}{\lambda_{n-p+1}}\right|$. Therefore the convergence can be very slow if the gap between $|\lambda_{n-p}|$ and $|\lambda_{n-p+1}|$ is not sufficiently large. Some techniques to speed up the convergence rate and save matrix-vector products involving $\mathcal{A}$ can be found in [86].

The main advantages of subspace iteration are its robustness and simple implementation. On the other hand, the linear rate of convergence makes it generally less attractive than the algorithms working with subspaces of increasing dimensions and projections, for example, the Arnoldi (or Lanczos) method [72, 44]. As introduced in the previous section, these Krylov subspace methods usually provide good approximations that may converge superlinearly to eigenvectors corresponding to extremal eigenvalues. However, as these methods proceed, both CPU time and storage needed to manage the Krylov subspace and invoke the Rayleigh-Ritz procedure increase. To keep the computational costs under control, various restarting strategies have been developed. Starting from the current perhaps large Krylov subspace, these strategies compute a new starting vector u_1 rich in the desired eigenvectors, from which a new Krylov subspace is generated. One of these restarting strategies, called *implicit restarting*, offers a particularly efficient and numerically stable formulation. The resulting algorithm is the well-known implicitly restarted Arnoldi method [85] described in Algorithm 2.5.

Algorithm 2.5 Implicitly restarted Arnoldi (IRA) method

```
Given a unit vector $u_1^{(0)}$, integers $k$ and $m$, compute a $m$-step Arnoldi decomposition $\mathcal{A}U_m^{(0)} = U_m^{(0)} H_m^{(0)} U_m^{(0)} + h_{m+1,m}^{(0)} u_m^{(0)} e_m^T$
for $i = 0, 1, ...,$ until convergence do
  Compute $\lambda(H_m^{(i)}) = \{\mu_1, \mu_2, ..., \mu_m\}$ and select $m-k$ shifts, for example, $\mu_{k+1}, \mu_{k+2}, ..., \mu_m$; set $Q_m^{(i)} \leftarrow I$
  for $j = k+1, ..., m$ do
    QR factorize $H_m^{(i)} - \mu_j I = Q_j R_j$;
    $H_m^{(i)} \leftarrow R_j Q_j + \mu_j I = Q_j^* H_m^{(i)} Q_j$;   $Q_m^{(i)} \leftarrow Q_m^{(i)} Q_j$;
  end for
  $U_{k+1}^{(i+1)} \leftarrow U_m^{(i)} Q_m^{(i)}(:, 1:k+1)$;  $H_k^{(i+1)} \leftarrow H_m^{(i)}(1:k, 1:k)$;
  $\tilde{h}_{k+1,k}^{(i)} \leftarrow H_m^{(i)}(k+1, k)$;   $q_{mk}^{(i)} \leftarrow Q_m^{(i)}(m, k)$;
  $\hat{u}_{k+1}^{(i+1)} \leftarrow \tilde{h}_{k+1,k}^{(i)} u_{k+1}^{(i)} + (h_{m+1,m}^{(i)} q_{mk}^{(i)}) u_{m+1}^{(i)}$;
  $h_{k+1,k}^{(i+1)} \leftarrow \|\hat{u}_{k+1}^{(i)}\|$;   $u_{k+1}^{(i+1)} \leftarrow \hat{u}_{k+1}^{(i+1)} / h_{k+1,k}^{(i)}$
  Beginning with the restarted $k$-step Arnoldi decomposition $\mathcal{A}U_k^{(i+1)} = U_k^{(i+1)} H_k^{(i+1)} + h_{k+1,k}^{(i+1)} u_k^{(i+1)} e_k^T$, perform $m-k$ Arnoldi steps to get $\mathcal{A}U_m^{(i+1)} = U_m^{(i+1)} H_m^{(i+1)} + h_{m+1,m}^{(i+1)} u_m^{(i+1)} e_m^T$ and test for convergence
end for
```

In the first step of this algorithm, the shifts $\{\mu_{k+1}, ...\mu_m\} \subset \lambda(H_m)$ consist of the set of unwanted Ritz values, whereas $\{\mu_1, ..., \mu_k\}$ is the set of wanted ones which are approximations to the user's desired eigenvalues, for example, those with smallest real parts or largest magnitude. This choice of Ritz values is called the *exact shifts* strategy, the default configuration in ARPACK [49]. Other shift strategies use harmonic Ritz values, or the roots of certain Chebyshev polynomials, Leja polynomials and least squares polynomials; see [3] and references therein.

It can be shown that the result of performing this computation corresponds to an implicit construction of the new k-dimensional Krylov subspace with starting vector

$u_1^{(i+1)} = (\mathcal{A} - \mu_{k+1}I)...(\mathcal{A} - \mu_m I)u_1^{(i)}$ up to a scaling factor. That is, the unwanted approximate eigenvector components corresponding to $\{\mu_{k+1}, ...\mu_m\}$ are filtered out by the implicit application of the filter polynomial. Therefore the new starting vector $u_1^{(i+1)}$ is expected to be richer in wanted eigenvectors than $u_1^{(i)}$.

Convergence analysis of IRA can be found in [85, 47, 4]. In practice, this algorithm generally converges superlinearly to the desired eigenpairs .

2.4 Spectral transformations

In many applications, the desired eigenvalues may not be well separated or are located in the interior of the convex hull of eigenvalues. Iterative eigenvalue methods introduced in the previous section usually have difficulties converging to these eigenvalues, because they provide approximations quickly only to well-separated extremal eigenvalues. In these situations, it is necessary to use a spectral transformation to map these eigenvalues to well-separated ones of a transformed eigenvalue problem. After the eigenpairs of the transformed problems are computed, the eigenvalues are transformed back to those of the original problem.

For the generalized eigenvalue problem $Av = \lambda Bv$, the shift-invert and the generalized Cayley transformation (see [53]) are as follows:

$$\begin{aligned} Av = \lambda Bv \quad &\Leftrightarrow \quad (A - \sigma_1 B)^{-1} Bv = \frac{1}{\lambda - \sigma_1} v \quad \text{(shift-invert)} \\ Av = \lambda Bv \quad &\Leftrightarrow \quad (A - \sigma_1 B)^{-1}(A - \sigma_2 B)v = \frac{\lambda - \sigma_2}{\lambda - \sigma_1} v \quad \text{(generalized Cayley).} \end{aligned} \tag{2.10}$$

It is easy to see that the shift-invert operator $\mathcal{A}_S = (A - \sigma_1 B)^{-1} B$ maps eigenvalues of $Av = \lambda Bv$ closest to σ_1 to eigenvalues of largest magnitude of $\mathcal{A}_S v = \mu v$ and those far from σ_1 to eigenvalues of small magnitude. The Cayley operator $\mathcal{A}_C = (A - \sigma_1 B)^{-1}(A - \sigma_2 B)$ maps the line $\Re(\lambda) = r$ $(r \neq \sigma_1)$ to a circle with center $1 + \frac{\sigma_1 - \sigma_2}{2(r - \sigma_1)}$ and radius $\frac{\sigma_1 - \sigma_2}{2(r - \sigma_1)}$; in particular, it maps eigenvalues of $Av = \lambda Bv$ to the right of $\Re(\lambda) = \frac{\sigma_1+\sigma_2}{2}$ to eigenvalues of $\mathcal{A}_C v = \mu v$ outside the unit circle and those to the left of this line to ones inside the unit circle (assuming that $\sigma_1 > \sigma_2$).

There is a simple connection between the shift-invert and Cayley transformations: $\mathcal{A}_C = I + (\sigma_1 - \sigma_2)\mathcal{A}_S$. Therefore, for a given starting vector u_1, the Krylov subspaces $\mathcal{K}_k(\mathcal{A}_S, u_1) = \mathcal{K}_k(\mathcal{A}_C, u_1)$ (see Section 2.2.1). Suppose the ordinary Arnoldi method (without restarting) is applied to $\mathcal{A}_S$ and $\mathcal{A}_C$ respectively for m steps with the same starting vector u_1. The resulting Arnoldi decompositions are $\mathcal{A}_S U_m^S = U_m^S H_m^S + h_{m+1,m}^S u_{m+1}^S e_m^T$ and $\mathcal{A}_C U_m^C = U_m^C H_m^C + h_{m+1,m}^C u_{m+1}^C e_m^T$, where $u_1^S = u_1^C = u_1$, the approximate eigenpairs are $(\mu_j^S, U_m^S w_j^S)$ and $(\mu_j^C, U_m^C w_j^C)$, with (μ_j^S, w_j^S) and (μ_j^C, w_j^C) being the eigenpairs of H_m^S and H_m^C respectively $(1 \le j \le m)$. Then we have the following lemma which can be derived immediately from Lemma 2.5 in [53].

Lemma 2.4.1 *Given the above Arnoldi decompositions, the two sets of approximate eigenpairs can be ordered such that* $\mu_j^C = 1 + (\sigma_1 - \sigma_2)\mu_j^S$ *and* $U_m^C w_j^C = U_m^S w_j^S$.

In other words, with the same staring vector u_1, there is no essential difference between $\mathcal{A}_S$ and $\mathcal{A}_C$ for the ordinary Arnoldi method. However, it should be made clear that the two operators do make a difference for the implicitly restarted Arnoldi (IRA) method. Specifically, assume we have the above two Arnoldi decompositions at the end of the first IRA cycle. Then for the restart, IRA chooses $m-k$ members of smallest magnitude from $\{\mu_j^S\}$ and $\{\mu_j^C\}$ respectively as shifts for the two operators. However, given the ordering of the two sets of approximate eigenvalues in Lemma 2.4.1, it is obvious that the shifts for $\mathcal{A}_S$ do *not* correspond to those for $\mathcal{A}_C$. In other words, the eigenvector components filtered out for the two operators during the restart are different. Consequently, for the restarted (the second) IRA cycle, the two k-dimensional subspaces spanned by the new Arnoldi vectors $U_k^{S(2)}$ and $U_k^{C(2)}$ are not the same; each subspace is rich in eigenvectors corresponding to dominant eigenvalues of its associated transformation operator.

In each step (outer iteration) of eigenvalue algorithms for the transformed problem, one or a few matrix-vector products involving the transformation operator need to be computed. This requires the solution of linear systems of the form $(A - \sigma B)x = y$. Traditionally (before the late-1990s), it was recommended that this solve be done using either sparse direct solvers based on factorizations of $A - \sigma B$ (whenever possible) or iterative solvers with tolerances slightly smaller than those used for the stopping criterion of the eigenvalue algorithms. When the matrices are so large that the solution by direct solvers becomes infeasible, iterative linear solvers (inner iteration) must be used in this setting to solve the linear systems to prescribed tolerances. This provides the prospect of inexact eigenvalue algorithms with "inner-outer" structure. The major focus of this thesis is to study some existing techniques and various new strategies to improve the effectiveness of several inexact eigenvalue algorithms when they are used to solve large sparse eigenvalue problems with spectral transformations.

2.5 Preconditioned Krylov subspace linear solvers

The most well known family of general purpose sparse iterative linear solvers is based on Krylov subspaces. Given a linear system of equations $Gz = b$ of order n and a starting vector $z_0 \in \mathbb{C}^n$ for the solution, Krylov subspace methods compute an approximate solution $z_k = z_0 + d_k$ in the k-th iteration, where d_k belongs to the *Krylov subspace*

$$\mathcal{K}_k(G, r_0) = \text{span}\{r_0, Gr_0, ..., G^{k-1}r_0\}, \tag{2.11}$$

where $r_0 = b - Gz_0$ is the initial residual vector. Since any vector in $\mathcal{K}_k(G, r_0)$ can be written as a polynomial in G times r_0, $d_k = \sum_{j=0}^{k-1} a_j G^j r_0 = \mathbf{q}_{k-1}(G)r_0$ for some polynomial $\mathbf{q}_{k-1}(t) = \sum_{j=0}^{k-1} a_j t^j$ of degree no greater than $k-1$. We measure the convergence of Krylov subspace methods through the residual vector

$$r_k = b - Gz_k = b - G(z_0 + d_k) = r_0 - G\mathbf{q}_{k-1}(G)r_0 = \mathbf{p}_k(G)r_0, \tag{2.12}$$

where the *residual polynomial* $\mathbf{p}_k(t) = 1 - t\mathbf{q}_{k-1}(t)$ with degree no greater than k satisfies $\mathbf{p}_k(0) = 1$. Different Krylov subspace methods compute d_k by choosing different polynomials $\mathbf{q}_{k-1}(t)$.

For a Hermitian positive definite G, the conjugate gradient method (CG) [39] is the most frequently used algorithm. Let $z = G^{-1}b$ be the true solution of the linear system, and $(z - z_k, z - z_k)_G^{1/2} = \|z - z_k\|_G$ be the G-norm of the error. It can be shown that z_k generated by the conjugate gradient method is the unique member of the translated Krylov space $z_0 + \mathcal{K}_k(G, r_0)$ for which the G-norm of the error is minimized, or alternatively, the Galerkin condition

$$r_k \perp \mathcal{K}_k(G, r_0) \tag{2.13}$$

is satisfied. The conjugate gradient method satisfies the following inequality

$$\|z - z_k\|_G \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|z - z_0\|_G, \tag{2.14}$$

where $\kappa = \frac{\lambda_{max}(G)}{\lambda_{min}(G)}$ is the condition number of G. This means the convergence of CG tends to be faster for smaller condition number of the coefficient matrix.

For a Hermitian indefinite G, the minimum residual method (MINRES) [64] computes the optimal approximate solutions in the translated Krylov space $z_0 + \mathcal{K}_k(G, r_0)$ for each k, in the sense that it minimizes the 2-norm of the residual vector

$$\|r_k\| = \min_{\substack{\mathbf{p}_k \in \mathbf{P}_k \\ \mathbf{p}_k(0)=1}} \|\mathbf{p}_k(G) r_0\| = \min_{\substack{z_k = z_0 + d_k \\ d_k \in \mathcal{K}_k(G, r_0)}} \|b - G z_k\|. \tag{2.15}$$

Here $\mathbf{P}_k$ is the set of all polynomials with degree no greater than k. This minimum residual property is equivalent to the Petrov-Galerkin condition that

$$r_k \perp G\mathcal{K}_k(G, r_0). \tag{2.16}$$

Note that the MINRES residual norm $\|r_k\|$ monotonically decreases with k as the minimizer z_k is taken from a larger translated Krylov subspace. It is shown in [36] that

$$\|r_k\| \leq 2 \left(\sqrt{\frac{\kappa - 1}{\kappa + 1}} \right)^{k-1} \|r_0\|, \tag{2.17}$$

where $\kappa = \frac{\max\{|\lambda(G)|\}}{\min\{|\lambda(G)|\}}$.

Both CG and MINRES are closely related to the Lanczos method introduced in Section 2.2.1 for estimating extremal eigenvalues of Hermitian matrices. The computational work of both linear solvers in each iteration are fixed: the conjugate gradient method requires one matrix-vector product, two inner products and three vector updates, and MINRES only needs two additional vector updates. In addition, the conjugate gradient iterate can be recovered easily from the MINRES iterate. See Chapter 2 of [19].

If the coefficient matrix G is non-Hermitian, the generalized minimum residual method (GMRES) [73] is the standard approach for constructing iterates satisfying the optimality condition same as that of MINRES for Hermitian G. In fact, the minimum residual property and Petrov-Galerkin condition for MINRES apply verbatim to GMRES. Roughly speaking, the extension of MINRES to GMRES can be derived by replacing the Lanczos method with the Arnoldi method. The convergence theory of GMRES is more complicated than those of CG and MINRES due to the geometry of the eigenvectors of non-Hermitian G. Specifically, let $G = V \Lambda V^{-1}$ where Λ is the diagonal matrix of eigenvalues of G. It can be shown that

$$\frac{\|r_k\|}{\|r_0\|} \leq \|V\| \|V^{-1}\| \min_{\substack{\mathbf{p}_k \in \mathbf{P}_k \\ \mathbf{p}_k(0)=1}} \max_{\lambda_j} |\mathbf{p}_k(\lambda_j)|. \tag{2.18}$$

Here, the polynomial $\mathbf{p}_k$ satisfying the "minimax" condition does not have an explicit form. In general, its asymptotic convergence can be bounded by the Chebyshev polynomial or Faber polynomial in the complex plane, as long as the numerical range $W(G) = \left\{ \frac{z^*Gz}{z^*z} : z \in \mathbb{C}^n, z \neq 0 \right\}$ does not include the origin; see [68, 21].

The work for orthogonalization of Arnoldi vectors and storage requirements of GMRES are proportional to kn at step k, and these costs become prohibitive for large k. A widely used approach to deal with this issue is to restart GMRES when k becomes large. In other words, an upper bound m on the dimension of the Krylov subspace is specified. If the residual norm has not decreased to the specified tolerance for $k \leq m$, then GMRES is stopped and restarted with z_k in place of z_0 as the initial iterate. This method is referred to as the GMRES(m) method. However, if the restart is invoked before the asymptotic convergence behavior of GMRES takes place, GMRES(m) may never converge at the asymptotic rate.

An alternative way for developing Krylov subspace linear solvers for non-Hermitian G is to force the computational costs in each iteration to be fixed. This can be achieved by a variant of the Lanczos algorithm based on biorthogonalization of two Krylov subspaces. This class of algorithms include the biconjugate gradient method (BICG) [23], the quasi-minimal residual (QMR) method [31], the transpose-free QMR [30], the biconjugate gradient stabilized method (BICGSTAB) [93] and BICGSTAB(l) [80]. In this thesis, we restrict our attention to MINRES and GMRES for the inner solves.

When applied to the linear systems $Gz = b$ directly, Krylov subspace linear solvers are likely to suffer from slow convergence and lack of robustness. In these situations, preconditioning is a key strategy for improving performance. The basic idea of preconditioning is to transform the original linear system by a linear process into a new system with the same solution, but that is likely to be easier to solve with an iterative linear solver. For instance, a preconditioning matrix P can be found and used to transform $Gz = b$ to $GP^{-1}\tilde{z} = b$ with $z = P^{-1}\tilde{z}$. It is well known that the convergence of Krylov subspace methods strongly depends on the eigenvalue distribution of the (preconditioned) coefficient matrix GP^{-1}. If P is a good approximation to G in some sense, the iterative solvers applied to $GP^{-1}\tilde{z} = b$ are expected to converge much more quickly because the eigenvalues of GP^{-1} tend to be well-clustered. For this idea to work, it is necessary that application of the action of P^{-1}, i.e., solution of a linear system of the form $Px = y$, be inexpensive. This condition guarantees that the cost of the preconditioned matrix-vector product involving GP^{-1} is not prohibitive.

The preconditioning process can be defined in many different ways. Some widely used preconditioning techniques include diagonal scaling, Jacobi, Gauss-Seidel and successive overrelaxation (SSOR), incomplete LU factorizations, approximate inverse and polynomial preconditioners, domain decomposition, multigrid, and many strategies which exploit the structure and spectral properties of the coefficient matrix (for example, preconditioners developed for G with specific saddle-point structures arising in fluid dynamics and numerical optimization). The development and analysis of efficient preconditioning techniques are essential to the success of iterative solution of linear systems of equations. The quality of the preconditioner generally has much more impact on the reliability and effectiveness of iterative techniques than the particular choice of a Krylov subspace solver. In this thesis, we assume that a reasonably good preconditioner is already available for the iterative solution of linear systems arising in inexact eigenvalue algorithms.

2.6 Related work

The development and analysis of inexact algorithms for eigenvalue problems with spectral transformation have attracted considerable interest in the past decade. In this subsection, we briefly review the literature closely related to this thesis.

Inexact inverse iteration is the most simple inexact eigenvalue algorithm and the best understood one. Early references [33, 43] establish the linear convergence of the outer iteration for non-Hermitian problems, assuming that the algorithm uses a fixed shift and a sequence of *decreasing* tolerances for the solution of the linear systems. Inexact Rayleigh quotient iteration for symmetric matrices is studied in [84] and [58], where the authors explore how the error of the solution to the linear systems affects the convergence of the outer iteration. Systematic analysis of this algorithm is given by Spence and his collaborators; see [5, 6, 7, 26, 25]. A major concern in these papers is the connection between the error of the inner solve and the convergence of the outer iteration, with different choices of variable shifts, tolerances and formulations of the linear systems. Meanwhile, there has been increasing interest in reducing the inner iteration cost to enhance the effectiveness of the algorithm. Reference [75] gives some new insights into preconditioning the linear systems arising in inexact Rayleigh quotient iteration by modifying the right hand side of the preconditioned system. This idea is extended in [5, 6, 7] and further refined in [25, 26] for inexact inverse iteration or Rayleigh quotient iteration, where a special type of preconditioner with "tuning" is constructed and analyzed. We will introduce this idea in the next section.

Inexact subspace iteration is a straightforward block extension of inexact inverse iteration with a fixed shift. Reference [65] establishes linear convergence of the outer iteration of this algorithm for standard eigenvalue problems and show by the block-GMRES [68] convergence theory that tuning keeps the block-GMRES iteration counts roughly *constant* for solving the block linear systems, though the inner solve is required to be done with increasing accuracy as the outer iteration proceeds.

In all these algorithms, tuning makes the preconditioned right hand side of the linear system an approximate eigenvector (or invariant subspace) of the preconditioned system matrix, and hence the inner iteration counts are considerably reduced. Recently, it was found that the single and multiple vector iteration algorithms have a close connection with the simplified Jacobi-Davidson method; see [27, 29].

In the meantime, some developments have been made in understanding inexact projection-based eigenvalue algorithms, such as the Lanczos and the Arnoldi method.

Reference [8] carried out a large number of tests of the Arnoldi method and concluded that the matrix-vector product must be computed accurately in the initial Arnoldi steps, but the accuracy can be relaxed as the Arnoldi method proceeds without compromising the convergence of approximate eigenpairs. In other words, a sequence of *increasing* tolerances can be used for the solution of linear systems in the Arnoldi steps. This behavior was also discovered in [35] for the Lanczos method. Reference [74] used matrix perturbation theory to give an analysis of this observation for an inexact Arnoldi method. The use of inexact matrix-vector products has also been studied in the setting of Krylov subspace linear solvers; see [9, 10],[77, 78, 79] and [82, 92].

To further study the inexact Arnoldi method, [28] extends the tuning strategy and the relaxed accuracy of matrix-vector products to IRA with shift-invert transformation for standard eigenvalue problems. The authors construct tuning with Arnoldi vectors in the current IRA cycle, and show that an ILU preconditioner with tuning considerably reduces the inner iteration counts for a test problem from Matrix Market [51]. It is observed there and in Chapter 4 of the thesis that this improvement is mainly due to the fact that tuning helps cluster the eigenvalues of the preconditioned system matrix of the linear system in each Arnoldi step (outer iteration). In addition, [28] proposes a heuristic estimate of the allowable relaxed tolerances for the solution of the linear systems, using the distance between the spectra of two matrices containing the wanted and unwanted Ritz values to replace the separation between the two. Numerical experiments show that the combined use of tuning and relaxed tolerances greatly reduces the inner iteration counts.

Another important inexact eigenvalue algorithm developed recently is the shift-invert residual Arnoldi method (SIRA) [45, 46]. This method has a few similarities to the Jacobi-Davidson method: both aim at only one eigenpair at a time and expand the subspace with the solution to a correction equation where the right-hand side is the current eigenvalue residual vector (the coefficient matrices are different for the two methods, though); in the initial steps, the inner solves can be done with only a moderate accuracy, but the tolerance for the inner solves decreases as the outer iteration proceeds. A major difference between the two methods is that the Jacobi-Davidson method requires an orthogonal correction for the current approximate eigenvector. The convergence of the Jacobi-Davidson method is based on the analysis of Newton's method, whereas the SIRA method is studied through the classical analysis of Krylov subspaces. Reference [45] compared the SIRA method with the inexact Arnoldi method (without restart) to compute six smallest eigenvalues of a non-Hermitian problem of order 10000 and concluded that SIRA outperforms the inexact Arnoldi method. It would be interesting to carry out a complete theoretical and numerical comparison of SIRA with the inexact IRA method developed in this thesis.

2.7 Preconditioning with tuning

In this section, we give a unified description of a preconditioning technique used for the iterative solution of linear systems arising in all the three inexact eigenvalue algorithms studied in this thesis. This strategy, referred to as preconditioning with "tuning," was proposed and studied by Spence and his collaborators for various inexact eigenvalue algorithms. The key idea can be summarized as "making the right-hand side of the linear system an approximate eigenvector of the coefficient matrix with the tuned preconditioner". The effect of this approach is that a good approximate solution can be constructed in the first inner iteration by properly using previously solved linear systems.

In the setting of inexact eigenvalue algorithms, given a preconditioning matrix P for the inner solve, tuning chooses an appropriate orthonormal set of vectors X of small dimension and constructs a special low-rank modification $\mathbb{P}$ of P, such that

$\mathbb{P}X = AX$, where A comes from the eigenvalue problem $Av = \lambda v$ or $Av = \lambda Bv$. In other words, the tuned preconditioner $\mathbb{P}$ acts like A on X. For instance, it may be defined as $\mathbb{P} = P + (AX - PX)\left(X^*(A - P)X\right)^{-1}(AX - PX)^*$ for Hermitian A and P or $\mathbb{P} = P + (AX - PX)X^* = AXX^* + P(I - XX^*)$ for non-Hermitian A and P. Note that $\mathbb{P}X = AX$ is equivalent to $A\mathbb{P}^{-1}(AX) = AX$, i.e., the column vectors of AX span an invariant subspace of $A\mathbb{P}^{-1}$ with eigenvalue 1. In the following, let $\mathcal{X}$ be the space spanned by the columns of X.

The success of the tuning strategy depends on the proper choice of $\mathcal{X}$ for the linear systems arising in specific inexact eigenvalue algorithms. For all the three inexact eigenvalue algorithms studied in this thesis, given the linear system in a specified outer iteration, $\mathcal{X}$ is chosen to be the space spanned by the the solution vector(s) of the linear system(s) in the last one or few outer iterations. We show that with this choice of $\mathcal{X}$, the right-hand side of the linear system in the given outer iteration can be well approximated by vectors in $A\mathcal{X}$, and is therefore an approximate eigenvector of the coefficient matrix of this linear system with the tuned preconditioner $\mathbb{P}$. This idea will be explained in detail in later chapters.

3 Inexact Rayleigh quotient iteration

In this chapter, we present a detailed analysis of preconditioned MINRES for approximately solving the linear systems that arise when Rayleigh Quotient Iteration is used to compute the lowest eigenpair of a symmetric positive definite matrix. We provide some insights into the initial stagnation of MINRES iteration in both a qualitative and quantitative way, and show that both the asymptotic MINRES convergence rate and the rate at which MINRES iterate approximates the desired eigenvector mainly depend on how quickly the unique negative eigenvalue of the preconditioned shifted coefficient matrix is approximated by its corresponding harmonic Ritz value. By exploring when the negative Ritz value appears in MINRES iteration, we obtain a better understanding of the limitation of preconditioned MINRES in this context and the virtue of a new type of preconditioner with "tuning". Comparison of MINRES with SYMMLQ in this context is also given. Finally we show that tuning based on a rank-2 modification can be applied with little additional cost to guarantee positive definiteness of the tuned preconditioner.

3.1 Introduction

In this chapter, we study an inexact Rayleigh quotient iteration (RQI) for computing the lowest eigenpair of a Hermitian positive definite matrix A. RQI is one of the most basic eigenvalue algorithms, which refines the inverse power method by using the Rayleigh quotient σ as the shift to obtain increasingly accurate eigenpair estimates. Given a good starting eigenvector estimate, this algorithm converges cubically to the desired eigenpair if the shift-invert matrix-vector product $(A - \sigma I)^{-1}x$ in each (outer) iteration is computed exactly. This accurate computation is usually realized by solving $(A - \sigma I)y = x$ with a direct sparse linear solver. In this chapter, we are interested in the situation where the size of A is so large that direct solution is impractical, and thus this linear solve is carried out by some iterative methods (inner iteration), for example, by some variants of the MINRES algorithm. Specifically, we will investigate the long initial slow convergence period of MINRES iteration in this setting, analyze difficulties that arise when ordinary preconditioned MINRES is used for the inner solve, and provide new insights into a special type of preconditioner with tuning that avoids these difficulties and enhances the efficiency of the inner iteration.

For inexact Rayleigh quotient iteration where a Krylov subspace method is used for the inner solve, it is known that the linear solver inevitably suffers from a long period of initial slow convergence; that is, the residual norm of this linear system decreases very slowly in a large number of initial steps before it goes down at a reasonable steady rate during the asymptotic convergence period. This long initial latency is caused by the near singular coefficient matrix $A - \sigma I$ where the Rayleigh quotient σ is very close to the desired eigenvalue. Fortunately, reference [75] shows that the initial latency of the inner iteration may be accompanied by significant improvement of the eigenvector approximation by the inner iterate, and therefore the inner iteration can be terminated with a considerably improved eigenvector estimate before the linear residual norm becomes small enough. This perspective was also discussed in [58], where it was remarked that the linear residual norm is not a good measure of the error of inner solves for the purpose of eigenvalue computation. An alternative metric of the inner solve errors was proposed there to accurately evaluate the eigenvector improvement, and then was used to estimate the actual convergence rate of inexact RQI.

In this chapter, we first extend the observation in [75]. We give a qualitative description of the initial latency of unpreconditioned MINRES iteration in the setting of inexact RQI. Then, by analyzing unpreconditioned MINRES behavior in depth, we show *how quickly* the angle between the MINRES iterate and the desired eigenvector decreases as the MINRES iteration proceeds. We provide some evidence that the rate at which the MINRES iterate converge to the desired eigenvector is not affected by the initial latency of MINRES iteration; it only depends on an effective condition number of the shifted coefficient matrix. The analysis depends on the fact that the right-hand side of the unpreconditioned system is an approximate eigenvector of the coefficient matrix; it is based on the properties of harmonic Ritz values and their connection with the MINRES residual polynomial.

In practice, a symmetric positive definite preconditioner P need be used with MINRES for the inner solve. In this case, however, the right hand side of the preconditioned linear system is generally far from a good approximate eigenvector of the preconditioned coefficient matrix. The convergence theory of MINRES indicates that the inner iteration steps needed to reach a prescribed relative tolerance will grow significantly as the outer iteration proceeds. To resolve this difficulty, [75] introduces some new perspectives on preconditioning in this setting, namely, that faster convergence of inner iterations can be obtained by properly modifying the right hand side of the preconditioned linear system. However, cubic convergence rate of outer iterations is lost with this approach, because the linear system solved is different from that in RQI. This idea was refined in [26], where an alternative preconditioning approach called tuning is proposed and analyzed. Tuning defines a rank-1 modification of P which forces the tuned preconditioner $\mathbb{P}$ to behave in the same way as A on the current approximate eigenvector. It was shown there that tuning makes the right-hand side an approximate eigenvector of the preconditioned coefficient matrix; as a result, the inner iteration counts can be considerably reduced. In addition, cubic convergence of outer iterations can be retrieved, as tuning does not change the linear system.

To extend the study in [75, 26], we provide new insights into the limitations of preconditioning without tuning and show how tuning leads to a major improvement. Specifically, we show that without tuning, the starting Lanczos vector in the MIN-

RES iteration is far from a good approximate eigenvector, and therefore it takes large number of MINRES steps for the subspace for candidate solutions to be rich in the desired eigenvector. In addition, the number of these ineffective MINRES steps increases as the outer iteration proceeds. On the other hand, if MINRES with the tuned variant of P is used, considerably fewer MINRES steps are needed to get an obvious improvement of eigenvector approximation because the starting Lanczos vector is already close to the desired eigenvector. The analysis of unpreconditioned MINRES can be applied directly to the preconditioned MINRES with tuning. We then introduce a tuning strategy based on a rank-2 modification which avoids some potential numerical difficulties associated with the original tuning and guarantees positive definiteness of the tuned preconditioner.

This chapter is organized as follows. Section 3.2 reviews some preliminary facts for later discussions. Section 3.3 gives detailed convergence analysis of the inner iteration with zero starting vector for the three versions of MINRES and provides some comments on the different performance of MINRES and SYMMLQ in this setting. A rank-2 modification tuning is introduced in Section 3.4 as an improvement of the rank-1 modification tuning of [26]. Numerical experiments supporting the analysis are given in Section 3.5. Finally, we summarize the chapter in Section 3.7.

3.2 Preliminaries

We want to compute the lowest eigenpair of a symmetric positive definite matrix by Rayleigh Quotient Iteration. Consider the eigenvalue problem

$$Av = \lambda v, \tag{3.1}$$

where A is symmetric positive definite with eigenvalues $0 < \lambda_1 < \lambda_2 \leq ... \leq \lambda_n$. Let $V = [v_1, v_2, ..., v_n] = [v_1, V_2]$ be the matrix of orthonormal eigenvectors and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$ so that $V^T AV = \Lambda$. Algorithm 3.1 describes a typical version of inexact Rayleigh Quotient Iteration to find a simple eigenpair.

Algorithm 3.1 Inexact Rayleigh Quotient Iteration

```
Given $x^{(0)}$ with $\|x^{(0)}\| = 1$
**for** $i = 0, 1, ...,$ until convergence **do**
  1. Compute the Rayleigh Quotient $\sigma^{(i)} \leftarrow x^{(i)T} A x^{(i)}$
  2. Choose $\tau^{(i)}$ and solve $(A - \sigma^{(i)} I) y^{(i)} = x^{(i)}$ inexactly such that $\|x^{(i)} - (A - \sigma^{(i)} I) y^{(i)}\| \le \tau^{(i)}$
  3. Update $x^{(i+1)} \leftarrow y^{(i)} / \|y^{(i)}\|$ and test for convergence
**end for**
```

From here through the end of the chapter, we drop the superscripts (i) that denote the count of the outer iteration, because we are interested in the convergence of inner iterations. Suppose a normalized outer iterate x is close to v_1 such that

$$x = \sum_{k=1}^{n} v_k c_k = v_1 \cos\varphi + u \sin\varphi, \tag{3.2}$$

where u is a unit vector orthogonal to v_1; φ is the angle between x and v_1, so that $\cos\varphi = c_1 = v_1^T x$, and $\sin\varphi = \|[0, V_2]^T x\| = \sqrt{c_2^2 + \cdots + c_n^2}$ is small.

The Rayleigh quotient associated with x is

$$\sigma = x^T A x = c^T \Lambda c = \lambda_1 + \sum_{k=2}^{n} (\lambda_k - \lambda_1) c_k^2 = \lambda_1 + (\bar{\lambda} - \lambda_1) \sin^2\varphi, \tag{3.3}$$

where $\bar{\lambda} = \sum_{i=2}^{n} (\frac{c_i^2}{\sin^2\varphi}) \lambda_i \in [\lambda_2, \lambda_n]$ is a weighted average of $\lambda_2, ..., \lambda_n$ uniquely determined by u. Assume that λ_1 is well-separated from λ_2, and φ is so small that $|\lambda_1 - \sigma| = O(\sin^2\varphi) \ll |\lambda_2 - \sigma| = O(1)$; hence v_1 is the dominant eigenvector of $(A - \sigma I)^{-1}$, and the cubic convergence of RQI (see [62], p. 76) is easily established.

Recall that there is a connection between the Lanczos algorithm for eigenvalues of a symmetric matrix G and the MINRES and SYMMLQ methods for solving systems $Gy = b$. Given the starting vector $u_1 = b/\|b\|$, the Lanczos algorithm leads to

$$G U_m = U_m T_m + \beta_{m+1} u_{m+1} e_m^T = U_{m+1} \overline{T}_m \tag{3.4}$$

where the tridiagonal matrix $T_m = \text{tridiag}[\beta_j, \alpha_j, \beta_{j+1}]$ $(1 \leq j \leq m)$ comes from the well-known three-term recurrence formula. Our analysis mainly results from the convergence of the leftmost harmonic Ritz value to the leftmost eigenvalue of G, which depends on the approximation from the Krylov subspace range(U_m) to the associated eigenvector of G as m increases.

We will use a major theorem from [63], which characterizes the MINRES iterate and establishes a connection between the residual polynomial and the harmonic Ritz values. Our analysis builds on this theorem and the interlacing property of Ritz and harmonic Ritz values. We use MATLAB notation $w(1)$ to denote the first entry of a vector w.

Theorem 3.2.1 *Suppose* MINRES *is applied to solve the system $Gy = b$. At the m-th* MINRES *iteration step with the corresponding Lanczos decomposition in* (3.4), *the* MINRES *iterate is*

$$y_m = U_m M_m^{-2} T_m e_1 \beta_1, \tag{3.5}$$

where $M_m^2 = \overline{T}_m^T \overline{T}_m$, $\beta_1 = \|b\|$. The residual of the linear system is

$$r_m = b - Gy_m = U_{m+1} w w(1) \beta_1, \;\; \|r_m\| = |w(1)| \beta_1 \tag{3.6}$$

where $\|w\| = 1$, $w^T \overline{T}_m = 0^T$,

$$|w(1)| = \beta_{m+1} |f_m(1)| / (1 + \beta_{m+1}^2 \|f_m\|^2)^{1/2}, \tag{3.7}$$

and $f_m = T_m^{-1} e_m$. Moreover, the residual can be written as

$$r_m = \chi_m(G) b / \chi_m(0) \tag{3.8}$$

where $\chi(\lambda) = \prod_{i=1}^{m} (\lambda - \xi_i^{(m)}) = \det[\lambda I_m - T_m^{-1} M_m^2]$ is the residual polynomial whose *roots are the harmonic Ritz values* $\xi_i^{(m)}$*, defined as eigenvalues of the pencil* $M_m^2 - \xi T_m$*. It can be shown that* $GU_mM_m^{-1}$ *has orthonormal columns and* $1/\xi_i^{(m)}$ *are the eigenvalues of* $H_m = (GU_mM_m^{-1})^TG^{-1}(GU_mM_m^{-1}) = M_m^{-T}T_mM_m^{-1}$*.*

3.3 Convergence of MINRES in inexact RQI

In this section, we analyze the convergence of the three versions of MINRES for solving the linear system in RQI. We consider in turn unpreconditioned MINRES, preconditioned MINRES with an ordinary symmetric positive definite preconditioner P (without tuning), and preconditioned MINRES with a tuned variant of P. For all the three cases, we assume that MINRES iteration starts with a zero starting vector $y_0 = 0$.

The analysis is based on properties of harmonic Ritz values. To fix notation in the following subsections, we use θ for Ritz values, ξ for harmonic Ritz values, quantities with hat for the preconditioned system without tuning and those with tilde for the preconditioned system with tuning. G and b are respectively the shifted system matrix and right hand side of the (preconditioned) system in step 2 of Algorithm 3.1.

3.3.1 Unpreconditioned MINRES

It is observed in [75] that the convergence of unpreconditioned MINRES for $(A - \sigma I)y = x$ can be very slow when the Rayleigh quotient σ is close to λ_1, i.e., when $\varphi = \angle(x, v_1)$ is small enough. That is, the residual norm $\|r_m\| = \|x - (A - \sigma I)y_m\|$ remains still close to 1 for quite large m. We call this phenomenon initial slow convergence and describe it in the theorem below. To make the exposition smooth, we defer the proof to Section 3.6.1.

Theorem 3.3.1 *Suppose unpreconditioned* MINRES *is used to solve* $(A - \sigma I)y = x$ *in* RQI*, where* $x = v_1 \cos\varphi + u \sin\varphi$ *(see* (3.2)*). Assume that* u *has components of at least* m *eigenvectors of* A *so that* MINRES *will not give the exact solution at the* *first m steps. For any such fixed u, $\lim_{\varphi\to 0}\|r_k\| = 1$ for any $k \le m$. Moreover, for any given $k \le m$, if φ is small enough*[1]*, then $1 - \|r_k\| = O(\sin^2\varphi)$.*

Remark 3.1. This residual norm estimate shows qualitatively that the initial slow convergence of the inner iteration is more pronounced as the outer iterate x becomes closer to the true eigenvector v_1. For any given $k \le m$, the theorem shows that $\|r_k\|$ tends to be closer to 1 as φ becomes smaller.

In the context of using MINRES in RQI to compute (λ_1, v_1), we are more interested in how quickly $\angle(y_m, v_1)$ decreases with m. Theorem 4.1 of [75] establishes the fact that although the MINRES iteration appears to stagnate in its initial steps, $\angle(y_m, v_1)$ may decrease considerably during these iterations. We restate the theorem, and expand on the result by showing that the leftmost harmonic Ritz value $\xi_1^{(m)}$ plays a critical role in the behavior of $\angle(y_m, v_1)$.

Theorem 3.3.2 *Let (μ_i, v_i) be the eigenpairs of the shifted matrix $G = A - \sigma I$, with eigenvalues ordered as $0 < |\mu_1| < |\mu_2| \le \ldots \le |\mu_n|$. Let x be a unit norm approximation to v_1 with small $\varphi = \angle(x, v_1)$. Let y_m be the* MINRES *approximate solution in $\mathcal{K}_m(G, x)$ and $r_m = x - Gy_m = p_m(G)x$ be the associated linear residual. If $|p_m(\mu_1)| < 1$, then*

$$\tan\angle(y_m, v_1) \le \frac{|\mu_1|}{|\mu_2|}\frac{1}{|1 - p_m(\mu_1)|}\left(1 + \frac{(\|r_m\|^2 - |p_m(\mu_1)|^2\cos^2\varphi)^{1/2}}{\sin\varphi}\right)\tan\varphi \tag{3.9}$$

or approximately,

$$\tan\angle(y_m, v_1) \le \frac{|\xi_1^{(m)}|}{|\mu_2|}(1 + \max_{2\le i\le n}|p_m(\mu_i)|)\tan\varphi \tag{3.10}$$

Proof The result (3.9) is established in [75]. For (3.10), first recall that as φ is small, $G = A - \sigma I$ has only one negative eigenvalue $\mu_1 = \lambda_1 - \sigma = O(\sin^2\varphi)$ and the smallest positive eigenvalue is $\mu_2 = \lambda_2 - \sigma = O(1)$. Recall also the interlacing property

[1] How small is small enough depends on k; for bigger k, this threshold tends to be smaller

mentioned in [63], that the Ritz values $\{\theta_k^{(m)}\}$ interlace the harmonic Ritz values $\{\xi_k^{(m)}\} \cup \{0\}$. Since $\det[T_2] = -\beta_2^2 = \theta_1^{(2)}\theta_2^{(2)} < 0$, we have $\xi_1^{(2)} < \theta_1^{(2)} < 0 < \theta_2^{(2)} < \xi_2^{(2)}$. To analyze the convergence of MINRES, recall from Theorem 3.2.1 that the harmonic Ritz values $\xi_k^{(m)}$ are zeros of the residual polynomial $p_m(G) = \chi_m(G)/\chi_m(0)$. That is,

$$p_m(\mu) = \prod_{k=1}^{m}(1 - \mu/\xi_k^{(m)}). \tag{3.11}$$

Therefore, the residual vector can be represented as

$$\begin{aligned} r_m &= p_m(G)x = p_m(G)\sum_{i=1}^{n} c_i v_i = \sum_{i=1}^{n} p_m(\mu_i)c_i v_i \\ &= p_m(\mu_1)\cos\varphi\ v_1 + \sin\varphi \sum_{i=2}^{n}(c_i/\sin\varphi)p_m(\mu_i)v_i \\ &= \cos\varphi \prod_{k=1}^{m}(1-\mu_1/\xi_k^{(m)})v_1 + \sin\varphi \sum_{i=2}^{n}\omega_i \prod_{k=1}^{m}(1-\mu_i/\xi_k^{(m)})v_i, \end{aligned} \tag{3.12}$$

where $\mu_i = \lambda_i - \sigma$ and $\omega_i = c_i/\sin\varphi$ is such that $\sum_{i=2}^{n}\omega_i^2 = 1$. As $\sin\varphi$ is small and $\cos\varphi \approx 1$, it is clear that to make $\|r_m\|$ small, $p_m(\mu_1) = \prod_{k=1}^{m}(1-\mu_1/\xi_k^{(m)})$, the product of m factors, has to be small. This condition is satisfied if and only if the first factor $(1-\mu_1/\xi_1^{(m)})$ is small, because the product of the second through the m-th factor is slightly bigger than 1. In fact, as $\mu_1 < 0$ and $\xi_k^{(m)} > 0$ $(k = 2, ..., n)$,

$$1 < \prod_{k=2}^{m}(1-\mu_1/\xi_k^{(m)}) \approx 1 - \sum_{k=2}^{m}\mu_1/\xi_k^{(m)} < 1 + (m-1)|\mu_1|/\mu_2 = 1 + O(\sin^2\varphi). \tag{3.13}$$

Here we use the first order approximation of the product based on the facts that $\mu_1/\mu_2 = O(\sin^2\varphi) \ll 1$ and, from the interlacing property, that $\xi_2^{(m)}$ approximates μ_2 from above as m increases.

To get the new bound in (3.10), we need to estimate $\|r_m\|^2 - |p_m(\mu_1)|^2\cos^2\varphi$ and

$|1 - p_m(\mu_1)|$ in (3.9). Since $\{v_i\}$ are orthonormal, we know from (3.12) that

$$\|r_m\|^2 - |p_m(\mu_1)|^2 \cos^2 \varphi = \sin^2 \varphi \sum_{i=2}^{n} \left(\frac{c_i}{\sin \varphi} \right)^2 p_m(\mu_i)^2 \leq \sin^2 \varphi \max_{2 \leq i \leq n} p_m(\mu_i)^2, \tag{3.14}$$

where the inequality comes from the relation $\sum_{i=2}^{n} (c_i / \sin \varphi)^2 = 1$. The estimation of $|1 - p_m(\mu_1)|$ can be simplified using (3.13):

$$|1 - p_m(\mu_1)| = |1 - \prod_{k=1}^{m} (1 - \mu_1 / \xi_k^{(m)})| \approx |1 - (1 - \mu_1 / \xi_1^{(m)})| = |\mu_1 / \xi_1^{(m)}|. \tag{3.15}$$

The new bound (3.10) is easily established from the above two estimates and (3.9). ∎

Remark 3.2. The above theorem shows that, as also observed in [75], improvements of the approximate eigenvector can be obtained during the period of initial slow convergence of MINRES. In fact, since $p_m(\mu_1)(\cos \varphi) v_1$ is the dominant term in r_m (see (3.12)), MINRES is almost stagnant when $p_m(\mu_1)$ stays close to 1 during the initial steps. However, in such a scenario, $|1 - p_m(\mu_1)| \approx |\mu_1 / \xi_1^{(m)}|$ may be increasing from a minuscule number (say, 10^{-10}) to a moderately small number (say, 10^{-3} or 10^{-2}). As this quantity appears in the denominator in (3.9), $\tan \angle(y_m, v_1)$ may decrease significantly even though the MINRES residual remains close to 1.

Remark 3.3. Note that $\max_{2 \leq i \leq n} |p_m(\mu_i)|$ in (3.10) might not have significant effect on the behavior of $\angle(y_m, v_1)$ when m is not too large. Intuitively, if G has a wide spectrum (which is often the case if it is unpreconditioned), $\max_{2 \leq i \leq n} |p_m(\mu_i)|$ does not decrease considerably for small and moderate m since many eigenvalues μ_i cannot indeed be approximated by any harmonic Ritz value $\xi_k^{(m)}$; it becomes small only when m is large enough so that each eigenvalue μ_i is well-approximated by some harmonic Ritz value. Therefore, $\angle(y_m, v_1)$ decreases with m mainly because $\xi_1^{(m)}$ approximates $\mu_1 < 0$ from below ($|\xi_1^{(m)}|$ decreases to $|\mu_1|$). The behavior of MINRES for $Gy = x$ and the decrease of $\angle(y_m, v_1)$ both depend on how quickly $\xi_1^{(m)}$ approches μ_1.

To explore this point, we need to use the relation between Ritz values and the reciprocals of harmonic Ritz values. It is shown in Section 5 of [63] that for a Lanczos decomposition in (3.4), the reciprocals of the harmonic Ritz values of G are Ritz values of G^{-1} from an orthonormal basis of range(GU_m). Hence the convergence of $\xi^{(m)}$ to μ_1 depends on the convergence of the extreme Ritz value $1/\xi_1^{(m)}$ of G^{-1} to the corresponding eigenvalue $1/\mu_1$, which in turn depends on the convergence of angles between the Krylov subspace range(GU_m) and the eigenvector v_1 of G^{-1} associated with $1/\mu_1$. Since the columns of GU_m form a basis of $G\mathcal{K}_m(G,x)$, when $\angle(v_1, G\mathcal{K}_m(G,x))$ is small, the eigenvalue $1/\mu_1$ of G^{-1} can be well-approximated by the extreme Ritz value of G^{-1}, namely $1/\xi_1^{(m)}$, obtained from an orthonormal basis of $G\mathcal{K}_m(G,x) = \mathcal{K}_m(G,Gx)$.

The following two lemmas from Chapter 4 of [86] show the quality of the approximation from $G\mathcal{K}_m(G,x)$ to v_1, and lead to our main theorem, which describes how quickly $\xi_1^{(m)}$ approximates μ_1 as MINRES iteration proceeds.

Lemma 3.3.3 *Suppose G is symmetric and has an orthonormal system of eigenpairs (μ_i, v_i), with its eigenvalues ordered so that $\mu_1 < \mu_2 \leq \cdots \leq \mu_n$. Then*

$$\tan\angle(v_1, \mathcal{K}_k(G,u)) \leq \frac{\tan\angle(v_1,u)}{c_{k-1}(1+2\eta)}, \quad \text{where } \eta = \frac{\mu_1-\mu_n}{\mu_n-\mu_2} < -1. \tag{3.16}$$

Here $c_k(1+2\eta) = (1+2\sqrt{\eta+\eta^2})^k + (1+2\sqrt{\eta+\eta^2})^{-k}$ is the k-th order Chebyshev polynomial of the first kind for $|1+2\eta| > 1$.

Lemma 3.3.4 *Let (λ, v) be an eigenpair of a symmetric matrix C. Suppose U_φ is a set of orthonormal column vectors for which $\varphi = \angle(v, \text{range}(U_\varphi))$ is small. Then the Rayleigh quotient $H_\varphi = U_\varphi^T C U_\varphi$ has an eigenvalue λ_φ such that $|\lambda - \lambda_\varphi| \leq \|E_\varphi\|$, where $\|E_\varphi\| \leq \frac{\sin\varphi}{\sqrt{1-\sin^2\varphi}}\|C\| = \tan\varphi\|C\|$.*

Let $u = Gx$ in Lemma 3.3.3 and $C = G^{-1}$ in Lemma 3.3.4. Recalling that μ_1 is the eigenvalue of G closest to zero so that $\|G^{-1}\| = 1/|\mu_1|$, we immediately have the following main theorem. This theorem shows how quickly the leftmost harmonic Ritz value $\xi_1^{(m)}$ approaches the corresponding eigenvalue $\mu_1 < 0$ from below as the MINRES iteration proceeds. This result can be applied to (3.10) in Theorem 3.3.2 to show how quickly $\tan\angle(y_m, v_1)$ decreases.

Theorem 3.3.5 *Suppose unpreconditioned* MINRES *is used to solve $Gy = x$ in Rayleigh Quotient Iteration where $G = A - \sigma I$ and x is defined in* (3.2)*. Let $\xi_1^{(m)}$ be the leftmost (also the unique negative) harmonic Ritz value. Then*

$$\frac{1}{\xi_1^{(m)}} - \frac{1}{\mu_1} \leq \frac{1}{|\mu_1|}\frac{\tan\angle(v_1, Gx)}{c_{m-1}(1+2\eta)}, \quad \text{i.e.,} \quad 1 - \frac{\mu_1}{\xi_1^{(m)}} \leq \frac{\tan\angle(v_1, Gx)}{c_{m-1}(1+2\eta)}. \tag{3.17}$$

Suppose m_0 (depending on φ and η) is the smallest integer for which the second upper bound in (3.17) is smaller than 1. Note that as $\xi_1^{(m)} < \mu_1 < 0$ and $1-\mu_1/\xi_1^{(m)} < 1$ for all m, (3.17) holds trivially for $m < m_0$ because the upper bound is not smaller than 1. Therefore this theorem describes how quickly $\xi_1^{(m)}$ approaches μ_1 from below ($|\xi_1^{(m)}|$ decreases to $|\mu_1|$) for $m \geq m_0$. It provides insight into MINRES convergence and also sheds some light on the behavior of $\tan\angle(y_m, v_1)$ ($m \geq m_0$) described by (3.10) in Theorem 3.3.2. These points are elaborated on as follows.

We first analyze the numerator of the upper bound to explore MINRES convergence. Note that $Gx = (A - \sigma I)\sum_{i=1}^{n} c_i v_i = (\lambda_1 - \sigma)\cos\varphi v_1 + \sum_{i=2}^{i=n}(\lambda_i - \sigma)c_i v_i$, and

$$\begin{aligned} \tan\angle(v_1, Gx) &= \frac{\|[(\lambda_2-\sigma)c_2, ..., (\lambda_n-\sigma)c_n]\|}{|(\lambda_1-\sigma)\cos\varphi|} \\ &\leq \frac{(\lambda_n-\sigma)\sin\varphi}{O(\sin^2\varphi\cos\varphi)} = O\left(\frac{1}{\sin\varphi\cos\varphi}\right). \end{aligned} \tag{3.18}$$

Therefore, for a fixed η, as the outer iteration proceeds and x becomes closer to v_1 (φ becomes smaller), (3.12), (3.17) and (3.18) indicate that m_0 becomes bigger and more MINRES iterations are needed to make $\xi_1^{(m)}$ close to μ_1 and $1 - \mu_1/\xi_1^{(m)}$ obviously smaller than 1. Hence, MINRES suffers a longer initial slow convergence period, as it takes more iterations to significantly reduce the dominant component v_1 in r_m.

To see how rapidly $\tan\angle(y_m, v_1)$ and $\|r_m\|$ decrease for $m \geq m_0$, note that the denominator of the upper bound behaves like $(1 + 2\sqrt{\eta + \eta^2})^{m-1}$ asymptotically (Lemma 3.3.3). Hence we define $(1 + 2\sqrt{\eta + \eta^2})^{-1}$ as the *asymptotic convergence factor* (smaller than 1). Given φ, as bigger $|\eta|$ corresponds to smaller asymptotic convergence factor and smaller m_0, we expect faster convergence of $\xi_1^{(m)}$ to μ_1 for $m \geq m_0$. This indicates that $\tan\angle(y_m, v_1)$ decreases with m ($m \geq m_0$) more quickly and in addition, MINRES will converge more quickly after its initial slow convergence period.

Though (3.17) holds trivially for $m < m_0$, one may speculate that $\tan\angle(y_m, v_1)$ still decreases at a rate controlled by η in the initial MINRES steps. This speculation is corroborated to some extent by the following arguments. Reference [58] analyzes the case where the conjugate gradient (CG) method is used to perform the the system solve required by the Jacobi-Davidson method, and shows that the convergence of CG for the correction equation simply depends on the effective condition number of $(I - xx^T)(A - \sigma I)(I - xx^T)$, which is essentially the reduced condition number of $A - \sigma I$. On the other hand, [75] shows that when solving $(A - \sigma I)y = x$, Jacobi-Davidson with CG delivers the same inner iterate (up to a constant) as SYMMLQ. This result is extended in [27] for the *preconditioned* solve of non-Hermitian systems, when tuning is used for the full orthogonalization method (FOM). It can be shown readily that preconditioned SYMMLQ with tuning is equivalent to Jacobi-Davidson with preconditioned CG. Our numerical experiments in Section 5 show that when tuning is used (to make the preconditioned solve behave qualitatively like the unpreconditioned solve), the eigenvalue residual curves of the MINRES and SYMMLQ iterates usually go hand in hand. Thus it is reasonable to conclude that $\tan\angle(y_m, v_1)$ decreases at a rate only depending on η. As the numerical experiments show, though φ is quite small in the last outer iteration, the eigenvalue residual of the inner iterate still decreases at a reasonable rate in the initial MINRES steps with tuning.

One caveat mentioned in Chapter 4 in [86] is that the bound of angles in (3.16) might be far from sharp when the algebraically smallest eigenvalues of G are clustered together so that $|\eta|$ could be very close to 1, whereas the actual convergence of the angles might be much faster. Nonetheless, bigger $|\eta|$ is still a reliable predictor of faster convergence of $\xi_1^{(m)}$. In fact, η is closely related to the reduced condition number $\kappa = \mu_n/\mu_2$ of the coefficient matrix since $|\eta| = |\frac{\mu_1-\mu_n}{\mu_n-\mu_2}| = 1 + \frac{\mu_2-\mu_1}{\mu_n-\mu_2}$, and

$$1 + \frac{1}{\kappa - 1} = 1 + \frac{\mu_2}{\mu_n - \mu_2} < |\eta| < 1 + \frac{2\mu_2}{\mu_n - \mu_2} = 1 + \frac{2}{\kappa - 1}. \tag{3.19}$$

Hence smaller κ corresponds to bigger $|\eta|$ and smaller asymptotic convergence factor, and is helpful to make $1 - \mu_1/\xi_1^{(m)}$ decrease to 0 more rapidly. This agrees with the result in [26] that smaller κ tends to make MINRES converge more quickly.

We end this subsection with a comment on the assumption in Theorem 3.3.2 that $p_m(\mu_1) < 1$, which might not always be true for small m. However, this has minimal impact on our convergence analysis. Section 3.6.2 gives some details on this.

3.3.2 Preconditioned MINRES with no tuning

It is observed in [75] and [25] that solving $(A-\sigma I)y = x$ by MINRES with a symmetric positive definite preconditioner is considerably slower than one might expect based on performance of such preconditioners in the usual setting of linear system solution.

More specifically, let $P \approx A$ be some symmetric positive definite preconditioner of A, for example, an incomplete Cholesky factorization. We then need to solve

$$\hat{G}\hat{y} \equiv L^{-1}(A - \sigma I)L^{-T}\hat{y} = L^{-1}x, \tag{3.20}$$

where $\hat{y} = L^T y$ and $LL^T = P$. Let $\hat{\mu}_1 < 0$ be the eigenvalue of $\hat{G}$ closest to zero and $\hat{v}_1$ be the corresponding eigenvector. It follows from (3.11) that a necessary condition of MINRES convergence for the preconditioned system is that for any nonnegligible eigenvector component in the right hand side, the corresponding eigenvalue must be well-approximated by some harmonic Ritz value. Though the right hand side $L^{-1}x$ is not close to $\hat{v}_1$, it usually still has a large component of $\hat{v}_1$. Therefore, it is possible to eliminate the component of $\hat{v}_1$ in $\hat{r}_m$ (hence making $\|\hat{r}_m\|$ small enough) only if the leftmost harmonic Ritz value $\hat{\xi}_1^{(m)}$ approximates $\hat{\mu}_1 < 0$ well enough. However, the following theorem suggests that the number of MINRES steps required for this good approximation to appear tends to increase as the outer iteration proceeds with $\hat{G}$ becoming more nearly singular.

Theorem 3.3.6 *Consider the preconditioned system $\hat{G}\hat{y} \equiv L^{-1}(A-\sigma I)L^{-T}\hat{y} = L^{-1}x$ arising in RQI. Let the eigenvalues of $\hat{G}$ be ordered as $\hat{\mu}_1 < \hat{\mu}_2 \leq ... \leq \hat{\mu}_n$, and let the m-step Lanczos decomposition be $\hat{G}\hat{U}_m = \hat{U}_m\hat{T}_m + \hat{\beta}_{j+1}\hat{u}_{j+1}e_j^T$. Then a necessary condition for $\hat{T}_m$ to be indefinite is satisfied, if*

$$(3.21) \qquad m \geq \frac{\log(\sqrt{\hat{\mu}_n/|\hat{\mu}_1|}\tan\angle(\hat{v}_1, L^{-1}x))}{\log(1+2\sqrt{\hat{\eta}+\hat{\eta}^2})} + 1, \quad \text{where } \hat{\eta} = \frac{\hat{\mu}_1 - \hat{\mu}_n}{\hat{\mu}_n - \hat{\mu}_2}.$$

Proof Recall that the eigenvalues of $G = A - \sigma I$ satisfy $\mu_1 < 0 < \mu_2$, and by the Sylvester inertia law for $\hat{G} = L^{-1}GL^{-T}$, we have $\hat{\mu}_1 < 0 < \hat{\mu}_2$. Using the eigendecompositions $\hat{G} = \hat{V}\text{diag}(\hat{\mu}_1, ..., \hat{\mu}_n)\hat{V}^T$ and $\hat{T}_m = \hat{S}_m\hat{\Theta}_m\hat{S}_m^T = \hat{U}_m^T\hat{G}\hat{U}_m$, [63] shows that

$$(3.22) \qquad \hat{\Theta}_m = (\hat{U}_m\hat{S}_m)^T\hat{G}(\hat{U}_m\hat{S}_m) = \hat{W}_m^T\text{diag}(\hat{\mu}_1, ..., \hat{\mu}_n)\hat{W}_m$$

where $\hat{W}_m = \hat{V}^T\hat{U}_m\hat{S}_m$ has orthonormal columns. In other words, the Ritz value $\hat{\theta}$ is a weighted average of the eigenvalues $\hat{\mu}_i$ (see Section 5 of [63]).

To see the condition for $\hat{T}_m$ being indefinite, we need to explore if $\hat{v}_1$ can be well-represented in $\hat{W}_m$ so that $\hat{\mu}_1 < 0$ can be well-approximated by $\hat{\theta}_1^{(m)}$. Consider any, say, the i-th, column of $\hat{U}_m\hat{S}_m$: $t_i = \hat{U}_m\hat{S}_m e_i = \hat{v}_1\cos\psi + \hat{u}\sin\psi \in \text{range}(\hat{U}_m)$, where $\psi \geq \angle(\hat{v}_1, \text{range}(\hat{U}_m))$ (recall that $\angle(\hat{v}_1, \text{range}(\hat{U}_m))$ is the smallest angle between $\hat{v}_1$

and any vector in range($\hat{U}_m$)), $\hat{u} \in \operatorname{span}\{\hat{v}_2, ..., \hat{v}_n\}$ and $\|\hat{u}\| = 1$. Then

$$\begin{aligned} \hat{\theta}_i^{(m)} &= (\hat{V}^T t_i)^T \operatorname{diag}(\hat{\mu}_1, ..., \hat{\mu}_n)(\hat{V}^T t_i) \quad (1 \le i \le m) \\ &= (\cos\psi e_1 + \sin\psi e_1^{\perp})^T \operatorname{diag}(\hat{\mu}_1, ..., \hat{\mu}_n)(\cos\psi e_1 + \sin\psi e_1^{\perp}) \\ &= \hat{\mu}_1 \cos^2\psi + \hat{\mu}^* \sin^2\psi, \end{aligned} \tag{3.23}$$

where $e_1 = [1, 0, ..., 0]^T$, $\|e_1^{\perp}\| = 1$, and $\hat{\mu}^* = (e_1^{\perp})^T \operatorname{diag}(\hat{\mu}_1, ..., \hat{\mu}_n)(e_1^{\perp}) \in [\hat{\mu}_2, \hat{\mu}_n]$. Hence all Ritz values are positive if and only if $\tan^2\psi > |\hat{\mu}_1|/\hat{\mu}^*$. It follows that, since $\psi \ge \angle(\hat{v}_1, \operatorname{range}(\hat{U}_m))$, $\hat{T}_m$ is positive definite if $\tan^2\angle(\hat{v}_1, \operatorname{range}(\hat{U}_m)) > |\hat{\mu}_1|/\hat{\mu}^*$.

Therefore $\tan^2\angle(\hat{v}_1, \operatorname{range}(\hat{U}_m)) < |\hat{\mu}_1|/\hat{\mu}^*$ is a necessary condition to make $\hat{T}_m$ indefinite (hence $\theta_1^{(m)} < 0$). By Lemma 3.2, since

$$\tan\angle(\hat{v}_1, \operatorname{range}(\hat{U}_m)) < \frac{\tan\angle(\hat{v}_1, L^{-1}x)}{c_{m-1}(1+2\hat{\eta})} < \frac{\tan\angle(\hat{v}_1, L^{-1}x)}{(1+2\sqrt{\hat{\eta}+\hat{\eta}^2})^{m-1}}, \tag{3.24}$$

the necessary condition holds if the last term in (3.24) is smaller than $\sqrt{|\hat{\mu}_1|/\hat{\mu}_n}$. The conclusion follows by taking the logarithm of both sides. ∎

Remark 3.5. This theorem simply suggests that during the initial steps of preconditioned MINRES, the leftmost harmonic Ritz value $\hat{\xi}_1^{(m)}$ will not approximate the negative eigenvalue $\hat{\mu}_1$ of $\hat{G}$, and therefore $\|\hat{r}_m\|$ will not be greatly reduced. In fact, as $\hat{T}_m$ is positive definite for small m, it follows that $\hat{\xi}_1^{(m)} > \hat{\mu}_2 > 0$, by the property of harmonic Ritz values. Therefore (3.12) implies that the component $\hat{v}_1$ in $\hat{r}_m$ is indeed magnified, since all factors of $\prod_{k=1}^{m}(1 - \hat{\mu}_1/\hat{\xi}_k^{(m)})$ are bigger than 1. It is hence impossible for MINRES to perform well during these iterations.

Note that (3.10) in Theorem 3.3.2 cannot be used here to describe $\tan\angle(y_m, v_1)$ where $y_m = L^{-T}\hat{y}_m$ is the recovered iterate from preconditioned MINRES iterate. This is because the right hand side of (3.20) is in general far from an approximation of $\hat{v}_1$, and there is no obvious relation between the eigenpair of $\hat{G}$ and that of G. Our numerical experiments in Section 3.5 also suggest that no significant improvement of eigenvector approximation can be obtained during the initial MINRES iterations. In the next subsection, we show how tuning solves this difficulty and makes Theorem 3.3.2 applicable to the preconditioned system.

In addition, the number of the initial "bad" MINRES steps tends to grow as the outer iterate becomes closer to the true eigenvector. In fact, it is shown in [5] (Theorem 9.1) that $\hat{\mu}_1 = (\lambda_1 - \sigma)/\|Lv_1\|^2 + O((\lambda_1 - \sigma)^2) = O(\sin^2 \varphi)$. Since in general $\angle(\hat{v}_1, L^{-1}x) = O(1)$, the bound of m given in the above theorem is like $\log |\frac{C}{\sin\varphi}| / \log(1 + 2\sqrt{\hat{\eta} + \hat{\eta}^2})$, which increases as the outer iteration proceeds. This estimate of the number of bad MINRES steps clearly shows a major limitation of preconditioned MINRES without tuning when it is used in the setting of RQI. This insight is supported by numerical experiments in section 3.5.

3.3.3 Preconditioned MINRES with tuning

One way suggested in [75] to address the fact that preconditioning does not do as well as expected in this setting is to replace the preconditioned system $L^{-1}(A - \sigma I)L^{-T}\hat{y} = L^{-1}x$ by $L^{-1}(A - \sigma I)L^{-T}\hat{y} = L^T x$. This idea comes from the fact that the aim is not to accurately solve the original preconditioned system, but to make the eigenvalue residual associated with MINRES iterate decrease more quickly. The authors show that the modified right hand side $L^T x$ is close to the eigenvector of the system matrix corresponding to the negative eigenvalue and MINRES convergence can be considerably improved. References [34] and [56] also advocate the use of $L^T x$ as the starting vector of preconditioned Lanczos algorithm to compute a few eigenpairs of symmetric matrices. One needs to notice that the recovered MINRES iterate y_m in this case converges to $(A - \sigma I)^{-1} L L^T x$ instead of $(A - \sigma I)^{-1} x$. Though $(A - \sigma I)^{-1} L L^T x$ is not as good as $(A - \sigma I)^{-1} x$ to approximate v_1, it is in practice still better than x. This strategy works because y_m approximates $(A - \sigma I)^{-1} L L^T x$ so fast that for small and moderate m, it is a better approximation to v_1 than its counterpart obtained from the standard use of preconditioned MINRES for $(A - \sigma I)^{-1}x$, though the latter would win when m is large enough.

However, this method is not RQI iteration, and the cubic convergence of the outer iteration is lost. An alternative approach introduced in [26], known as "tuning", entails a rank-1 modification of the Cholesky factor L of the symmetric positive definite preconditioner $P = LL^T$ so that the tuned preconditioner $\mathbb{P} = \mathbb{L}\mathbb{L}^T$ satisfies $\mathbb{P}x = Ax$ (the construction of $\mathbb{L}$ is discussed in Section 3.4 below). The preconditioned system with tuning is thus

$$\tilde{G}\tilde{y} \equiv \mathbb{L}^{-1}(A - \sigma I)\mathbb{L}^{-T}\tilde{y} = \mathbb{L}^{-1}x, \tag{3.25}$$

leaving the RQI structure unchanged. Therefore the cubic convergence of the outer iteration is preserved.

Suppose $\tilde{v}_1$ is the eigenvector of $\tilde{G}$ corresponding to the eigenvalue $\tilde{\mu}_1 < 0$. There is a straightforward relation between $(\tilde{\mu}_1, \tilde{v}_1)$ and the eigenpair (μ_1, v_1) of $G = A - \sigma I$. Note that as $\mathbb{P}x = Ax$, $\mathbb{P}v_1 \approx Av_1$ as the RQI outer iterate $x \longrightarrow v_1$. It follows that

$$(A - \sigma I)v_1 = \mu_1 v_1 = \left(\frac{\mu_1}{\lambda_1}\right)\lambda_1 v_1 = \left(\frac{\mu_1}{\lambda_1}\right)Av_1 \approx \left(\frac{\mu_1}{\lambda_1}\right)\mathbb{P}v_1, \tag{3.26}$$

and hence

$$\mathbb{L}^{-1}(A - \sigma I)\mathbb{L}^{-T}(\mathbb{L}^T v_1) \approx \mathbb{L}^{-1}\left(\frac{\mu_1}{\lambda_1}\right)\mathbb{P}v_1 = \left(\frac{\mu_1}{\lambda_1}\right)(\mathbb{L}^T v_1). \tag{3.27}$$

The relation $(\tilde{\mu}_1, \tilde{v}_1) \approx (\mu_1/\lambda_1, \mathbb{L}^T v_1)$ is thus established (see Lemma 3.1 in [26]). Similarly, the right hand side of (3.25) is

$$\begin{aligned} \mathbb{L}^{-1}x \;&\approx\; \mathbb{L}^{-1}v_1 = \mathbb{L}^{-1}\left(\frac{1}{\lambda_1}\right)\lambda_1 v_1 = \left(\frac{1}{\lambda_1}\right)\mathbb{L}^{-1}Av_1 \\ &\approx\; \left(\frac{1}{\lambda_1}\right)\mathbb{L}^{-1}\mathbb{P}v_1 = \left(\frac{1}{\lambda_1}\right)\mathbb{L}^T v_1 \approx \left(\frac{1}{\lambda_1}\right)\mathbb{L}^T x. \end{aligned} \tag{3.28}$$

In other words, the right hand side of the preconditioned system with tuning automatically approximates the starting vector $\mathbb{L}^T x$ proposed by [34] and [56] in the preconditioned Lanczos method and by [75] in preconditioned MINRES used in the context of RQI. In addition, it is an approximate eigenvector of the system matrix corresponding to $\tilde{\mu}_1 < 0$. Recall that this is the case for the unpreconditioned system $(A - \sigma I)y = x$. In fact, it is shown in [26] that $\sin\tilde{\varphi} \equiv \sin\angle(\tilde{v}_1, \mathbb{L}^{-1}x) = O(\sin\varphi)$. Therefore, the analysis of unpreconditioned MINRES directly applies to (3.25). The Ritz value $\tilde{\theta}_1^{(m)}$ and harmonic Ritz value $\tilde{\xi}_1^{(m)}$ are negative at the very beginning of the MINRES iterations, as in the unpreconditioned case. Compared to preconditioned MINRES with *no* tuning, the overhead of performing "bad" MINRES iterations in which $\hat{\xi}_1^{(m)} > 0$ is avoided with the tuned preconditioner. As a result, MINRES begins to converge earlier and more importantly, $\tan\angle(y_m, v_1)$ (where $y_m = \mathbb{L}^{-T}\tilde{y}_m$) decreases much more rapidly in the initial steps. See the figures in Section 3.5.

Similar to the unpreconditioned case, the convergence of preconditioned MINRES with tuning and decrease of $\tan\angle(y_m, v_1)$ basically depends on how quickly $\tilde{\xi}_1^{(m)}$ approaches $\tilde{\mu}_1$ from below. We have the following bound just like (3.17):

$$(3.29) \qquad 1 - \frac{\tilde{\mu}_1}{\tilde{\xi}_m^{(1)}} \le \frac{\tan\angle(\tilde{v}_1, \tilde{G}\mathbb{L}^{-1}x)}{c_{k-1}(1 + 2\tilde{\eta})}, \quad \text{where } \tilde{\eta} = \frac{\tilde{\mu}_1 - \tilde{\mu}_n}{\tilde{\mu}_n - \tilde{\mu}_2} < -1.$$

Preconditioned MINRES with tuning converges much more quickly than unpreconditioned MINRES, because the asymptotic convergence factor of the former is considerably smaller than that of the latter. See section 3.5 for comparisons of the two quantities. Note that by definition, η of the unpreconditioned MINRES is a constant that only depends on the eigenvalues of A, whereas $\hat{\eta}$ and $\tilde{\eta}$ may change as the outer iteration proceeds; in our experience, these changes in the preconditioned eigenvalues tend to be small.

Preconditioned MINRES with tuning also has an initial slow convergence period if the outer iterate x is close to v_1. In Section 3.6.1 we show that the relative linear residual $\|\tilde{r}_m\|/\|\mathbb{L}^{-1}x\| = 1 - O(\sin^2\tilde{\varphi})$ holds in the same way as for the unpreconditioned MINRES solve. The initial slow convergence is less pronounced for the preconditioned case with tuning because its asymptotic convergence factor is smaller.

3.3.4 Comparison of SYMMLQ and MINRES used in RQI

To solve the linear systems arising in RQI, a natural alternative to MINRES is SYMMLQ. With extensive numerical tests, Dul in [16] claimed that MINRES improves eigenvector approximation to some prescribed level in considerably fewer iterations than SYMMLQ. Rigorous analysis and comparison of the two methods is not seen in the literature. Here we provide some comments on the two solvers in this context.

Our experience is that MINRES is better than SYMMLQ in general, but the advantage may vary considerably depending on the preconditioned problem. In one of our sample problems with appropriate tuned preconditioner, there is little difference between the two methods, but for ill-conditioned problems without a preconditioner, as shown in [16], SYMMLQ might not even be able to improve the eigenvalue residual in a reasonable number of iterations.

To compare the MINRES iterate y_m^{MR} and SYMMLQ iterate y_m^{SL}, we see that the MINRES linear residual for $Gy = x$ is $x - Gy_m^{MR} = p_m^{MR}(G)x$ (by the definition of the MINRES residual polynomial p_m), so that

$$\begin{aligned} y_m^{MR} &= G^{-1}(I - p_m^{MR}(G))x = (I - p_m^{MR}(G))(G^{-1}x) = (I - p_m^{MR}(G))\sum_{i=1}^{n} b_i v_i \\ &= \sum_{i=1}^{n}(1 - p_m^{MR}(\mu_i))b_i v_i = \sum_{i=1}^{n}\left(1 - \prod_{j=1}^{m}(1 - \mu_i/\xi_j^{(m)})\right)b_i v_i, \qquad (3.30) \end{aligned}$$

where $G^{-1}x = \sum_{i=1}^{n} b_i v_i$ is the true solution. Similarly for SYMMLQ, we have

$$y_m^{SL} = \sum_{i=1}^{n}(1 - p_m^{SL}(\mu_i))b_i v_i = \sum_{i=1}^{n}\left(1 - \prod_{j=1}^{m}(1 - \mu_i/\theta_j^{(m)})\right)b_i v_i. \qquad (3.31)$$

The above expressions show clearly that the difference between y_m^{MR} and y_m^{SL} as approximations to v_1 simply results from the different quality of approximation to the extreme eigenvalue μ_1 and the interior eigenvalues μ_i by harmonic Ritz and Ritz values. Since $\angle(y_m, v_1)$ largely depends on the ratio of the magnitudes of eigenvectors corresponding to interior eigenvalues to that of v_1 contained in y_m, we speculate that the reason for $\angle(y_m^{MR}, v_1) < \angle(y_m^{SL}, v_1)$ is that harmonic Ritz values tend to be better approximations to the interior eigenvalues, though μ_1 is better approximated by the Ritz value $\theta_1^{(m)}$ (see [54], [63] and [81]).

Reference [16] also shows that the curve of eigenvalue residuals of MINRES iterates is generally smooth, whereas that of SYMMLQ iterates tends to be oscillatory. This phenomenon can be explained qualitatively by the fact the interior eigenvalues are susceptible to being impersonated by non-converged Ritz values. That is, an interior eigenvalue μ_k can be well-approximated by some Ritz value at the m-th step of the Lanczos process when the angle between the eigenvector v_k and the current Krylov subspace range(U_m) is not small [86]. At the m-th SYMMLQ step, a small number of interior eigenvalues μ_k might be impersonated by some "incorrect" Ritz value $\theta_{j(k)}^{(m)}$ (the subscript $j(k)$ is a function of k; $1 < j(k) < m$) so that $1 - \mu_k/\theta_{j(k)}^{(m)}$ is fairly small, and hence $1 - \prod_{j=1}^{m}(1 - \mu_k/\theta_j^{(m)})$ decreases dramatically. But in the next SYMMLQ step the impersonation may disappear and this quantity recovers its magnitude in the step before impersonation. This causes $\angle(y_m^{SL}, v_1)$ to fluctuate considerably. MINRES does not have this problem, however: a harmonic Ritz value would not well approximate an eigenvalue unless the corresponding eigenvector is well-represented in range(U_m) (see page 293 of [86], equation (4.19)). As a result, $1 - \prod_{j=1}^{m}(1 - \mu_k/\xi_j^{(m)})$ will not fluctuate greatly as m increases, and the decreasing curve of eigenvalue residuals is smoother. We use this observation in Section 3.5 to develop stopping criteria for the inner iterations.

3.4 Preconditioner with tuning based on a rank-2 modification

The symmetric preconditioner with tuning defined in [26] is based on a rank-1 modification of the Cholesky factor L of the ordinary symmetric positive definite preconditioner $P = LL^T$. We restate Lemma 3.2 from [26] to construct the tuned Cholesky factor.

Lemma 3.4.1 *Suppose $P = LL^T \approx A$ is a symmetric positive definite preconditioner of A. Let x be an approximation of v_1 and $w = Ax - Px$. The tuned Cholesky factor $\mathbb{L}$ is defined as $\mathbb{L} = L + \alpha w(L^{-1}w)^T$, where α is the real solution of $(L^{-1}w)^T(L^{-1}w)\alpha^2 + 2\alpha - \frac{1}{w^Tx} = 0$. Then $\mathbb{L}\mathbb{L}^Tx = Ax$.*

The tuned preconditioner $\mathbb{P} = \mathbb{L}\mathbb{L}^T$ can also be defined equivalently as a symmetric rank-1 modification of P. In fact,

$$
\begin{aligned}
\mathbb{P} &= \mathbb{L}\mathbb{L}^T = (L + \alpha w(L^{-1}w)^T)(L + \alpha w(L^{-1}w)^T)^T \\
&= LL^T + 2\alpha ww^T + ((L^{-1}w)^T(L^{-1}w))\alpha^2 ww^T = P + \frac{ww^T}{w^Tx} \\
&= P + \frac{(Ax - Px)(Ax - Px)^T}{(Ax - Px)^Tx},
\end{aligned}
\tag{3.32}
$$

such that $\mathbb{P}x = Ax$. This definition has the advantage enabling $\mathbb{P}$ to be defined for preconditioners not specified by Cholesky factors.

The tuned preconditioner $\mathbb{P}$ has to be positive definite for MINRES. It is shown in [26] that two conditions must be satisfied to guarantee positive definiteness, namely

$$
(Ax - Px)^Tx \neq 0, \quad \textit{and} \quad 1 + \frac{(Ax - Px)^TQ^{-1}(Ax - Px)}{(Ax - Px)^Tx} \geq 0. \tag{3.33}
$$

However, it is possible that $(Ax - Px)^Tx$ is 0 or small enough to cause numerical problems. Moreover, it is shown in [26] that in cases where $(Ax - Px)^Tx < 0$, the second condition above is satisfied only if $\|A - P\|$ is very small. The latter requirement is difficult to enforce except in cases where the Cholesky factor is very dense; for example, P can be the incomplete Cholesky preconditioner with very small drop tolerance.

Positive definiteness of a tuned preconditioner can be enforced with less stringent constraints by using a rank-2 modification of P. This approach is used to construct approximate Hessians for quasi-Newton methods in optimization ([57], Ch 11). In particular, we can use the BFGS modification

$$\mathbb{P} = P - \frac{(Px)(Px)^T}{(Px)^T x} + \frac{(Ax)(Ax)^T}{(Ax)^T x}. \tag{3.34}$$

It is easy to see that $\mathbb{P}x = Ax$. Lemma 11.5 in [57] shows that if the denominator of the last term in (3.34) is positive (which is the case here), $\mathbb{P}$ is positive definite.

A tuned preconditioner based on the rank-2 modification is slightly more expensive to apply than that based on the rank-1 modification. One should try the rank-1 modification and turn to the rank-2 version only when the former is not positive definite, i.e., when there is no real solution to the equation in Lemma 3.4.1.

3.5 Numerical Experiments

We compare unpreconditioned MINRES, preconditioned MINRES without tuning, and preconditioned MINRES with tuning for solving the linear system in RQI, in numerical experiments on three benchmark eigenvalue problems from MatrixMarket [51]. The first problem $nos5.mtx$ is a real symmetric positive definite matrix of order 468 coming from finite element approximation to a biharmonic operator that describes beam bending in a building. The second consists of two matrices $K = bcsst\mathbf{k}08.mtx$ and $M = bcsst\mathbf{m}08.mtx$ of order 1074 that define a generalized symmetric positive definite eigenvalue problem $Kx = \lambda Mx$ used for dynamic modeling of a structure. This generalized problem can be easily transformed to the standard problem $M^{-1/2}KM^{-1/2}(M^{1/2}x) = \lambda(M^{1/2}x)$ where the coefficient matrix can be formed directly because M is a positive definite diagonal matrix. The last one is a generalized symmetric positive semi-definite problem of order 2003 from fluid flows defined by symmetric positive definite $K = bcsst\mathbf{k}13.mtx$ and symmetric positive semi-definite $M = bcsst\mathbf{m}13.mtx$ with rank 1241. The first two examples show the differences among the three versions of MINRES. The third problem suggests that tuning might be used for more complex eigenvalue problems.

3.5.1 Stopping criteria for inner iterations

Efficiency of each solver is evaluated by the MINRES iteration counts needed in a given outer iteration to satisfy some stopping criterion. Note that in MINRES iteration, we can easily monitor the SYMMLQ iterate also because it can be obtained for free [19]. We define $eigres_m^{MR}$ and $eigres_m^{SL}$ to be the eigenvalue residual associated with the MINRES iterate y_m^{MR} and the SYMMLQ iterate y_m^{SL} respectively, and we stop the MINRES iteration when the relative changes of $\|y_m^{MR}\|$, $eigres_m^{MR}$ and $eigres_m^{SL}$ are all small enough. In other words, the stopping criterion is

$$(3.35) \qquad \text{stop}(\|y_m\|) \ \& \ \text{stop}(eigres_m^{MR}) \ \& \ \text{stop}(eigres_m^{SL}),$$

where

$$(3.36) \qquad \text{stop}(\|y_m\|) \equiv \frac{|\ \|y_{m-k}\| - \|y_{m-k-1}\|\ |}{\|y_{m-k}\|} < \epsilon_{inner}, \quad k = 0, 1,$$

and $\text{stop}(eigres_m^{MR})$ and $\text{stop}(eigres_m^{MR})$ are defined similarly.

We elaborate on this strategy as follows: Our aim is to stop MINRES as soon as $\angle(y_m, v_1) \approx \angle(y_{exact}, v_1)$ (the cubic convergence of the outer iteration is thus preserved). The first criterion is adopted by [75], where it is shown to be roughly equivalent to the condition $\text{stop}(|1 - p_m(\mu_1)|)$. This is a necessary condition for $p_m(\mu_1) \ll 1$ (say $p_m(\mu_1)$ is of order 10^{-3} to 10^{-2}), which in turn implies that MINRES has started to converge. Our experience is that $\angle(y_m, v_1) \approx \angle(y_{exact}, v_1)$ usually holds when MINRES has started to converge. The second criterion is directly connected to the eigenvalue problem: since the right hand side is dominated by v_1, we expect $\angle(y_m, v_1) \approx \angle(y_{exact}, v_1)$ once the eigenvalue residual stops decreasing. However, with just these two criteria, MINRES might stop prematurely due to a possibly slow approximation process. The criterion stop($eigres_m^{SL}$) helps prevent an early stop, since $eigres_m^{SL}$ tends to be oscillatory until $\angle(y_m, v_1)$ approximates $\angle(y_{exact}, v_1)$ well (see Section 3.3.4), whereas in our experience, $eigres_m^{MR}$ tends to stagnate slightly before this (see Figures 3.1–3.3). Finally, we require the stopping criteria to be satisfied for two successive steps to further ensure that MINRES does not stop prematurely.

One could instead choose a smaller ϵ_{inner} and stop MINRES when the criteria are satisfied for only one step, but this usually makes MINRES continue for quite a few steps after $\angle(y_m, v_1) \approx \angle(y_{exact}, v_1)$. We take $\epsilon_{inner} = 0.01$ for all the criteria in the tests. The combined criteria guarantee a fair comparison of preconditioned MINRES without and with tuning for solving the linear systems in RQI.

Note that we choose not to use the residual of the linear system, $\|x-(A-\sigma I)y_m\|$, in the stopping criteria, because as Figures 3.1–3.3 show, it is not possible to specify an extent to which the norm of the linear residual should be decreased for all problems when $\angle(y_m, v_1) \approx \angle(y_{exact}, v_1)$ holds.

3.5.2 Results and comments

We use the incomplete Cholesky preconditioner from MATLAB 7.4 with drop tolerance 0.25 for problem 1 and 2, and 0.0015 for problem 3. In each test the starting vector $x^{(0)}$ is chosen to be close enough to the target eigenvector v_1 so that the Rayleigh quotient $\sigma^{(0)}$ satisfies $|\lambda_1 - \sigma^{(0)}| < |\lambda_2 - \sigma^{(0)}|$. The results for MINRES in the third outer iteration of RQI on these problems are shown in Figures 3.1–3.3 and Tables 3.1–3.3.

Tables 3.1–3.2 show clearly that unpreconditioned MINRES converges slowly; as shown in Section 3.3.2, this is because $\tan(v_1, Gx) = O(\frac{1}{\sin\varphi\cos\varphi})$, and the asymptotic convergence factor is very close to 1 (i.e., the reduced condition number is big); see (3.17) and (3.19). In fact, unpreconditioned MINRES fails to satisfy the stopping criteria in the specified maximum number of steps. From now on, we only compare the preconditioned MINRES without and with tuning.

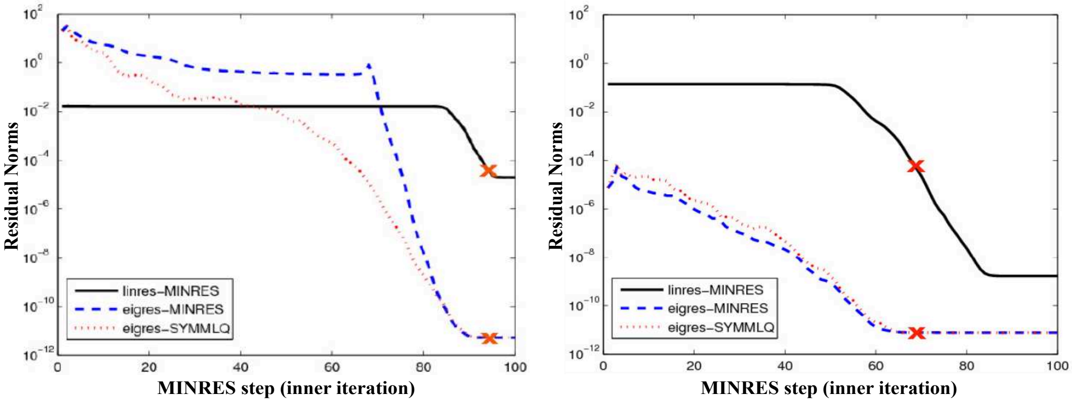

Figure 3.1: MINRES linear residual, MINRES and SYMMLQ eigenvalue residual in the third outer iteration on Problem 1, with drop tol 0.25. Left: preconditioned solve without tuning. Right: preconditioned solve with rank-1 tuning.

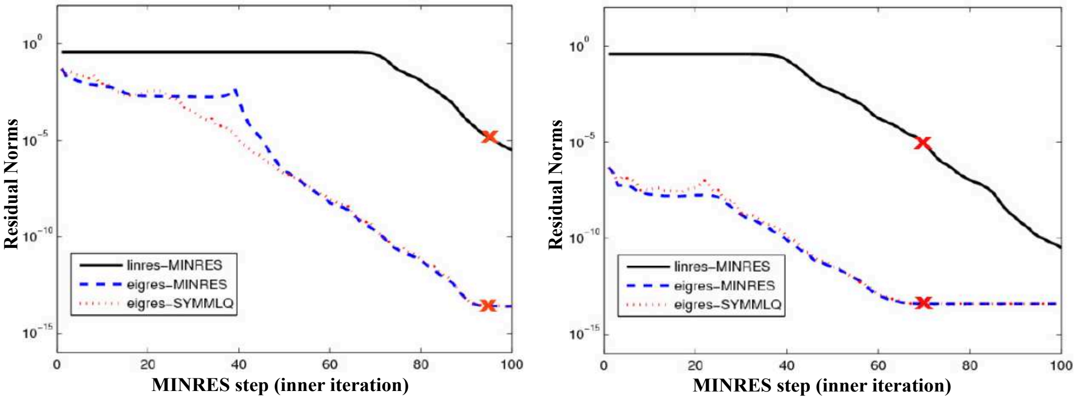

Figure 3.2: MINRES linear residual, MINRES and SYMMLQ eigenvalue residual in the third outer iteration on Problem 2, with drop tol 0.25. Left: preconditioned solve without tuning. Right: preconditioned solve with rank-2 tuning.

It is obvious from Figures 3.1–3.2 that preconditioned MINRES with tuning significantly outperforms the version without tuning. The cross marks on the curves indicate the MINRES iteration at which the stopping criteria are satisfied. It takes more steps for preconditioned MINRES without tuning to satisfy the stopping criteria than the version with tuning. The eigenvalue residual curve (dashed lines) of the tuned MINRES iterate is well below that of the untuned one, and the norm of the residual of the linear system (solid lines) also decreases more quickly due to tuning. Moreover, 1) the eigenvalue residual curve decreases slowly in the first dozens of steps of MINRES without tuning, and 2) the eigenvalue residual curve of preconditioned MINRES without tuning starts at a value much larger than the value at which the curve of the version with tuning starts.

	Non	No tuning	Tuning
MINRES iter	160*	94	68
neg Ritz shows in	2	64	1
aymptotic cvg. factor	0.9901	0.9189	0.9189
reduced cond. number	8.6172e+3	5.1497e+2	5.1509e+2
initial angle	3.6915e–3	3.6942e–1	3.9601e–5

Table 3.1: Comparison of three MINRES methods in the third outer iteration on Problem 1

	Non	No tuning	Tuning
MINRES iter	200*	95	69
neg Ritz shows in	2	31	1
aymptotic cvg. factor	0.9984	0.9347	0.9347
reduced cond. number	1.5154e+6	8.2201e+2	8.2201e+2
initial angle	1.5345e–4	2.3665e–3	1.1692e–6

Table 3.2: Comparison of three MINRES methods in the third outer iteration on Problem 2

Both the phenomena 1) and 2) can be explained by the fact that tuning forces the preconditioning operator to behave like A on the current outer iterate x. The reason for phenomenon 1) is given in Section 3.3.2: in the initial steps of MINRES without tuning, the negative eigenvalue of the preconditioned coefficient matrix cannot be approximated by any harmonic Ritz value because $\hat{T}_m$ is positive definite, and hence MINRES cannot perform well. Moreover, Table 3.3 shows that the number of these "bad" MINRES steps increases as the outer iteration proceeds, as Theorem 3.3.6

suggests. To explain phenomenon 2), first suppose $\hat{y}_0 = 0$ for the preconditioned MINRES without tuning. It follows that $\hat{y}_1 \in \hat{y}_0 + \mathcal{K}_1(\hat{G}, \hat{b})$ is a multiple of the preconditioned right hand side $\hat{b} = L^{-1}x$, and the recovered iterate $y_1 = L^{-T}\hat{y}_1$ is a multiple of $L^{-T}L^{-1}x = P^{-1}x$, which is in general far from a good approximation of v_1. Similarly for the preconditioned MINRES with tuning, y_1 is a multiple of $\mathbb{P}^{-1}x$. Since $\mathbb{P}$ and A behave in the same way on $x \approx v_1$, it is reasonable to expect that $\mathbb{P}^{-1}x \approx A^{-1}x \approx \lambda_1^{-1}v_1$, which is a much better approximation to v_1 than $\mathbb{P}^{-1}x$.

Tables 3.1–3.2 provide data supporting the above comparison. First, note that there is little difference in the asymptotic convergence factor and the reduced condition number between the preconditioned MINRES without and with tuning. The difference comes from the last rows in the two tables: the angle between the preconditioned right hand side and the eigenvector of the preconditioned coefficient matrix corresponding to the unique negative eigenvalue is much bigger in the case without tuning than it is in the case with tuning. As explained, it is this very fact that makes the first MINRES iterate with tuning ($\mathbb{P}^{-1}x$) a much better approximation to v_1 than that without tuning ($P^{-1}x$). Moreover, for the untuned preconditioner, $\hat{T}_m$ is positive definite in the first 63 steps in Problem 1 and in the first 30 steps in Problem 2. One can see from Figures 3.1–3.2 that the eigenvalue residual curves start to decrease quickly soon after $\hat{T}_m$ becomes indefinite.

We show by the third test that tuning can also be used for generalized eigenvalue problems that cannot be converted into standard eigenvalue problems. Since $M = bcsst\mathbf{m}13.mtx$ is singular, one has to solve $(K - \sigma M)y = Mx$ in Rayleigh Quotient Iteration. Similar to the previous standard problems, the tuned preconditioner $\mathbb{P}$ is a rank-1 modification of the preconditioner $P \approx K$ such that $\mathbb{P}x = Kx$. Our convergence analysis of MINRES may not be applied directly, because the eigenvectors are now M-orthogonal and expressions of the entries of the tridiagonal matrix T_m become less clear. Moreover, the fact that Mx is not close to the "negative"

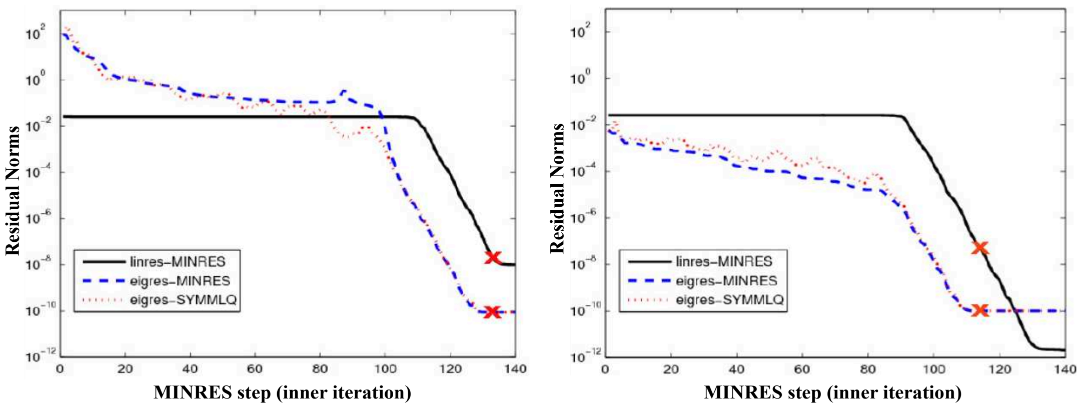

Figure 3.3: MINRES linear residual, MINRES and SYMMLQ eigenvalue residual in the third outer iteration on Problem 3, with drop tol 0.0015. Left: preconditioned solve without tuning. Right: preconditioned solve with rank-2 tuning.

eigenvector of $K - \sigma M$ makes the Ritz value analysis more complicated. However, Figure 3.3 and Table 3.3 show that the pattern observed in the previous two standard eigenvalue problems still holds for this problem.

Tables 3.4–3.5 show some cases when the rank-2 tuning has to be used. In problems 2 and 3, the rank-1 tuning makes the tuned preconditioner indefinite when the drop tolerance is above some threshold, and rank-2 tuning works with any drop tolerance. In the three test problems, there is little performance difference between preconditioned MINRES with the rank-1 and the rank-2 tuning. As the drop tolerance increases, the iteration counts of preconditioned MINRES with and without tuning both increase, but the difference between them becomes more pronounced.

3.6 Some technical details

3.6.1 Proof of initial slow convergence

3.6.1.1 Unpreconditioned MINRES

The proof of Theorem 3.3.1 is given as follows:

Proof Note from (3.2) that $\sin\varphi = \sqrt{c_2^2 + \cdots + c_n^2}$ and $u = \sum_{k=2}^{n} \frac{c_k}{\sqrt{c_2^2+\cdots+c_n^2}} v_k$. That is, u is uniquely determined by the ordered set $\{c_k/\sqrt{c_2^2+\cdots+c_n^2}\}$; one can fix u

Outer Iteration	1	2	3	4
Problem 1	7	19	64	
Problem 2	1	1	31	44
Problem 3	1	8	82	

Table 3.3: Numbers of preconditioned MINRES iteration steps without tuning needed to have $\theta_1^{(m)} < 0$

Drop tolerance	0.05	0.07	0.1	0.25	0.3	0.35
No Tuning	51	75	82	95	111	139
Rank-1 Tuning	35	51	60	–	–	–
Rank-2 Tuning	36	52	59	69	77	97

Table 3.4: Number of preconditioned MINRES iteration steps needed to satisfy the stopping criterion in the third outer iteration for Problem 2

Drop tolerance	2.5e–4	5.0e–4	7.5e–4	1.0e–3	1.25e–3	1.5e–3
No Tuning	76	84	103	112	122	133
Rank-1 Tuning	71	73	90	–	–	–
Rank-2 Tuning	65	73	89	99	107	115

Table 3.5: Number of preconditioned MINRES iteration steps needed to satisfy the stopping criterion in the third outer iteration for Problem 3

and only change φ by increasing/decreasing $\{c_k\}(2 \leq k \leq n)$ by a common factor, to see qualitatively how MINRES convergence is affected by φ.

One can see from (3.5) that $y_1 = 0$ since $T_1 = [0]$. We now assume that $m \geq 2$.

Recall the spectral decomposition of A, the Rayleigh quotient (3.3) and Lanczos decomposition (3.4). For any $k \le m$, we have

$$
\begin{aligned}
(3.37) \qquad x^T(A-\sigma I)^k x = c^T(\Lambda-\sigma I)^k c &= \sum_{i=2}^{n}(\lambda_i-\sigma)^k c_i^2 + (\lambda_1-\sigma)^k c_1^2 \\
&= \sin^2\varphi \sum_{i=2}^{n}\Big((\lambda_1-\sigma)+(\lambda_i-\lambda_1)\Big)^k \left(\frac{c_i^2}{\sin^2\varphi}\right) + (\lambda_1-\sigma)^k\cos^2\varphi \\
&= \sin^2\varphi \sum_{i=2}^{n}\Big(\sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix}(\lambda_1-\sigma)^j(\lambda_i-\lambda_1)^{k-j}\Big)\left(\frac{c_i^2}{\sin^2\varphi}\right) + (\lambda_1-\sigma)^k\cos^2\varphi \\
&= \sin^2\varphi \sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix}(\lambda_1-\sigma)^j\Big(\sum_{i=2}^{n}(\lambda_i-\lambda_1)^{k-j}\Big(\frac{c_i^2}{\sin^2\varphi}\Big)\Big) + (\lambda_1-\sigma)^k\cos^2\varphi.
\end{aligned}
$$

Letting $l_s = \sum_{i=2}^{n}(\lambda_i-\lambda_1)^s(\frac{c_i^2}{\sin^2\varphi}) \in [\lambda_2-\lambda_1, \lambda_n-\lambda_1]$ (which depends only on $\{\lambda_i\}$ and u, but not on φ) be a weighted average of $\{(\lambda_i-\lambda_1)^s\}(0 \le s \le k)$, and using $\lambda_1-\sigma = (\lambda_1-\bar{\lambda})\sin^2\varphi$, we then have

$$
\begin{aligned}
(3.38) \quad x^T(A-\sigma I)^k x &= \sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} l_{k-j}(\lambda_1-\bar{\lambda})^j\sin^{2j+2}\varphi + (\lambda_1-\bar{\lambda})^k\sin^{2k}\varphi\cos^2\varphi \\
&= l_k\sin^2\varphi + \begin{pmatrix} k \\ 1 \end{pmatrix} l_{k-1}(\lambda_1-\bar{\lambda})\sin^4\varphi + \cdots + \begin{pmatrix} k \\ k-1 \end{pmatrix} l_1(\lambda_1-\bar{\lambda})^{k-1}\sin^{2k}\varphi \\
&\quad +(\lambda_1-\bar{\lambda})^k\sin^{2k+2}\varphi + (\lambda_1-\bar{\lambda})^k\sin^{2k}\varphi\cos^2\varphi \\
&= \sin^2\varphi\Big(l_k + \cdots + \Big(\begin{pmatrix} k \\ k-1 \end{pmatrix} l_1(\lambda_1-\bar{\lambda})^{k-1} + (\lambda_1-\bar{\lambda})^k\Big)\sin^{2k-2}\varphi\Big) \\
&= q_{k-1}(\sin^2\varphi)\sin^2\varphi,
\end{aligned}
$$

where q_{k-1} is a polynomial of degree $k-1$ whose coefficients depend on $\{\lambda_i\}$ and u; in particular, l_k, the constant term of q_{k-1}, is independent of φ.

We can thus find the first few entries of T_m in closed form. For example,

$$
(3.39) \qquad \alpha_1 = x^T(A-\sigma I)x = \sigma - \sigma = 0,
$$

$$
\begin{aligned}
(3.40) \qquad \beta_2 &= \|(A-\sigma I)x - \alpha_1 x\| = \sqrt{x^T(A-\sigma I)^2 x} = \sqrt{q_1(\sin^2\varphi)\sin^2\varphi} \\
&= \sqrt{q_1(\sin^2\varphi)}\sin\varphi = \sqrt{l_2}\sin\varphi + O(\sin^3\varphi),
\end{aligned}
$$

$$
\begin{aligned}
(3.41) \qquad \alpha_2 &= u_2^T(A-\sigma I)u_2 = \left(\frac{(A-\sigma I)x}{\beta_2}\right)^T (A-\sigma I)\left(\frac{(A-\sigma I)x}{\beta_2}\right) \\
&= \frac{x^T(A-\sigma I)^3 x}{\beta_2^2} = \frac{q_2(\sin^2\varphi)\sin^2\varphi}{q_1(\sin^2\varphi)\sin^2\varphi} = \frac{q_2(\sin^2\varphi)}{q_1(\sin^2\varphi)},
\end{aligned}
$$

and

$$
\begin{aligned}
(3.42) \qquad \beta_3 &= |(A-\sigma I)u_2 - \beta_2 x - \alpha_2 u_2\| = \left\|\frac{(A-\sigma I)^2 x}{\beta_2} - \beta_2 x - \frac{\alpha_2(A-\sigma I)x}{\beta_2}\right\| \\
&= \sqrt{\frac{x^T(A-\sigma I)^4 x}{\beta_2^2} + \beta_2^2 + \alpha_2^2 - 2\beta_2^2 - 2\alpha_2^2 + 2\alpha_2\alpha_1} \\
&= \sqrt{\frac{x^T(A-\sigma I)^4 x}{x^T(A-\sigma I)^2 x} - \left(\frac{x^T(A-\sigma I)^3 x}{x^T(A-\sigma I)^2 x}\right)^2 - x^T(A-\sigma I)^2 x} \\
&= \sqrt{\frac{q_3(\sin^2\varphi)}{q_1(\sin^2\varphi)} - \left(\frac{q_2(\sin^2\varphi)}{q_1(\sin^2\varphi)}\right)^2 - q_1(\sin^2\varphi)\sin^2\varphi}
\end{aligned}
$$

One can similarly evaluate α_3 and β_4, though the expressions for other entries become much more complicated. Note that as the coefficients of q_{k-1} are uniquely determined by u, for any fixed u, all the entries in T_m are functions of $\sin^2\varphi$ only. (Obviously, if x consists of $m+1$ eigenvectors of A, then $(A-\sigma I)U_{m+1} = U_{m+1}T_{m+1}$ with $\beta_{m+2} = 0$, and $\|r_{m+1}\| = 0$. This is why we restrict our analysis to how $\|r_k\|$ $(1 \le k \le m)$ is affected as φ goes to zero.)

To show that $\lim_{\varphi\to 0}\|r_k\| = 1$ for all k with $1 \le k \le m$, we only need to establish the result for $k = m$, since the MINRES residual norm decreases monotonically. In light of Theorem 3.2.1, the key point is to show that $f_m(1)$ is the unique dominant entry in $f_m = T_m^{-1}e_m$. In fact, the entries of f_m can be evaluated by Cramer's rule as follows:

$$
\begin{aligned}
(3.43) \qquad |f_m(1)| &= \frac{|\det[e_m, T_m(1:m, 2:m)]|}{|\det[T_m]|} = \frac{\prod_{k=2}^{m} \beta_k}{\beta_2^2 \det[T_m(3:m, 3:m)]} \\
&= \frac{1}{\beta_2} \frac{\prod_{k=3}^{m} \beta_k}{|\det[T_m(3:m, 3:m)]|},
\end{aligned}
$$

and therefore

$$
(3.44) \qquad \frac{1}{|f_m(1)|^2} = \beta_2^2 \left(\frac{|\det[T_m(3:m, 3:m)]|}{\prod_{k=3}^{m} \beta_k} \right)^2,
$$

where $T_m(i:m, j:m)$ is the submatrix of T_m consisting of its ith through mth rows and jth through mth columns.

We now show that $\frac{1}{|f_m(1)|^2} = O(\sin^2 \varphi)$ and hence $\lim_{\varphi \to 0} \frac{1}{|f_m(1)|^2} = 0$. Simple observation from (3.38) shows that $\lim_{\varphi \to 0} q_{k-1}(\sin^2 \varphi) = l_k$ (the constant term of the polynomial), and thus the limit of α_2 and β_3 are l_3/l_2 and $\sqrt{l_4 l_2 - l_3^2}/l_2$ respectively. We can show by induction that all α_k and β_k have some *nonzero* limit (independent of φ) except for $\alpha_1 = 0$ and $\beta_2 \approx \sqrt{l_2} \sin \varphi$. Since all β_k $(3 \leq k \leq m)$ and all entries in $\det[T_m(3:m, 3:m)]|$ have a nonzero limit as $\varphi \to 0$, the term in (3.44) multiplying β_2^2 also has a nonzero limit. Recalling from (3.40) that $\beta_2^2 = q_1(\sin^2 \varphi) \sin^2 \varphi$, we have $\frac{1}{|f_m(1)|^2} = O(\sin^2 \varphi)$ and $\lim_{\varphi \to 0} \frac{1}{|f_m(1)|^2} = 0$.

Note that β_2 in the second column of T_m is the only nonzero entry of the first row, and hence $f_m(2) = 0$. Then

$$
\begin{aligned}
(3.45) \qquad |f_m(3)| = \frac{|\det[T_m(1:m, 1:2), e_m, T_m(1:m, 4:m)]|}{|\det[T_m]|} \\
= \frac{\beta_2^2 \prod_{k=4}^{m} \beta_k}{\beta_2^2 |\det[T_m(3:m, 3:m)]|} = \frac{\prod_{k=4}^{m} \beta_k}{|\det[T_m(3:m, 3:m)]|},
\end{aligned}
$$

Therefore, $|f_m(3)|$ has a nonzero limit as $\varphi \to 0$, and $\lim_{\varphi \to 0} \frac{f_m(3)^2}{f_m(1)^2} = 0$. One can show in the same way that $\lim_{\varphi\to 0}\frac{f_m(k)^2}{f_m(1)^2}=0$ $(4\le k\le m)$. Using (3.7), we have

$$\begin{aligned}\frac{1}{|w(1)|^2} &= \frac{1+\beta_{m+1}^2\|f_m\|^2}{\beta_{m+1}^2|f_m(1)|^2}=\frac{1}{\beta_{m+1}^2|f_m(1)|^2}+1+\sum_{k=2}^{m}\frac{f_m(k)^2}{f_m(1)^2}\\ &= 1+O(\sin^2\varphi)\end{aligned}\tag{3.46}$$

The assertion follows immediately from (3.6). ∎

3.6.1.2 Preconditioned MINRES with tuning

We can use the same reasoning to show that $\lim_{\varphi\to 0}\|\tilde{r}_m\|/\|\mathbb{L}^{-1}x\|=1$ for preconditioned MINRES with tuning.

Let $\tilde{G}\tilde{U}_m=\tilde{U}_m\tilde{T}_m+\tilde{\beta}_{j+1}\tilde{u}_{j+1}e_j^T$ be the m-step Lanczos decomposition. The first Lanczos vector $\tilde{u}_1=\mathbb{L}^{-1}x/\|\mathbb{L}^{-1}x\|=\tilde{v}_1\cos\tilde{\varphi}+\tilde{u}\sin\tilde{\varphi}$, where $\tilde{u}\perp\tilde{v}_1$ and $\|\tilde{u}\|=1$. Since the smallest eigenvalue of $\tilde{G}$ is $\tilde{\mu}_1=O(\sin^2\tilde{\varphi})$ (by Theorem 9.1 of [5]), we have the first entry of $\tilde{T}_m$ as follows:

$$\begin{aligned}\tilde{\alpha}_1 &= \tilde{u}_1^T\tilde{G}\tilde{u}_1=(\tilde{v}_1\cos\tilde{\varphi}+\tilde{u}\sin\tilde{\varphi})^T\tilde{G}(\tilde{v}_1\cos\tilde{\varphi}+\tilde{u}\sin\tilde{\varphi})\\ &= \tilde{\mu}_1\cos^2\tilde{\varphi}+\bar{\tilde{\mu}}\sin^2\tilde{\varphi}=O(\sin^2\tilde{\varphi}),\end{aligned}\tag{3.47}$$

where $\bar{\tilde{\mu}}\in[\tilde{\mu}_2,\tilde{\mu}_n]$. In light of (3.40) and (3.41), we can show easily that $\tilde{\beta}_2=O(\sin\tilde{\varphi})$, $\tilde{\alpha}_2$, $\tilde{\beta}_3$ and all other entries have a nonzero limit as $\tilde{\varphi}$ goes to zero (where we recall from the comment right after (3.25) that $\sin\tilde{\varphi}=O(\sin\varphi)$. Analysis of $\tilde{f}_m=\tilde{T}_m^{-1}e_m$ is similar to that of f_m, as follows:

$$\begin{aligned}|\tilde{f}_m(1)| &= \frac{|\det[e_m,\tilde{T}_m(1:m,2:m)]|}{|\det[\tilde{T}_m]|}\\ &= \frac{\prod_{k=2}^{m}\tilde{\beta}_k}{|\tilde{\alpha}_1\det[\tilde{T}_m(2:m,2:m)]-\tilde{\beta}_2^2\det[\tilde{T}_m(3:m,3:m)]|}\\ &= \frac{O(\sin\tilde{\varphi})}{O(\sin^2\tilde{\varphi})}=O\left(\frac{1}{\sin\tilde{\varphi}}\right)\end{aligned}\tag{3.48}$$

$$
\begin{aligned}
|\tilde{f}_m(2)| &= \frac{|\det[\tilde{T}_m(1:m,1), e_m, \tilde{T}_m(1:m,3:m)]|}{|\det[\tilde{T}_m]|} \\
&= \frac{|\tilde{\alpha}_1| \prod_{k=3}^m \tilde{\beta}_k}{|\tilde{\alpha}_1 \det[\tilde{T}_m(2:m,2:m)] - \tilde{\beta}_2^2 \det[\tilde{T}_m(3:m,3:m)]|} \\
&= \frac{O(\sin^2 \tilde{\varphi})}{O(\sin^2 \tilde{\varphi})} = O(1)
\end{aligned} \tag{3.49}
$$

One can show other entries of $\tilde{f}_m$ also have a nonzero limit, and hence $\tilde{f}_m(k)/\tilde{f}_m(1) = O(\sin \tilde{\varphi})(2 \leq k \leq n)$. Using exactly the same reasoning in Section 3.6.1.1, we can show that $\lim_{\tilde{\varphi}\to 0} 1/|\tilde{w}(1)|^2 = 1$ and the relative linear residual $\lim_{\tilde{\varphi}\to 0} \|\tilde{r}_m\|/\|\mathbb{L}^{-1}x\| = 1$. Similar to the unpreconditioned solve, if $\tilde{\varphi}$ is small enough, then $1 - \|\tilde{r}_m\|/\|\mathbb{L}^{-1}x\| = O(\sin^2 \tilde{\varphi})$.

3.6.2 Assumption of Theorem 3.3.2

In fact,

$$
\begin{aligned}
p_m(\mu_1) &= \prod_{k=1}^{m} (1 - \mu_1/\xi_k^{(m)}) \approx (1 - \mu_1/\xi_1^{(m)})(1 - \sum_{k=2}^{m} \mu_1/\xi_k^{(m)}) \\
&= 1 - \sum_{k=2}^{m} \mu_1/\xi_k^{(m)} - (\mu_1/\xi_1^{(m)})(1 - \sum_{k=2}^{m} \mu_1/\xi_k^{(m)}),
\end{aligned} \tag{3.50}
$$

which is smaller than 1 if and only if

$$
\xi_1^{(m)} > -\left(\frac{1 - \sum_{k=2}^m \mu_1/\xi_k^{(m)}}{\sum_{k=2}^m 1/\xi_k^{(m)}} \right). \tag{3.51}
$$

On the other hand, we can find the closed form of $\xi_1^{(2)}$ and $\xi_2^{(2)}$ by the definition of harmonic Ritz values. We do this by solving the generalized eigenvalue problem $M_2^2 w = \xi T_2 w$, where, by Theorem 3.2.1,

$$
M_2^2 = \bar{T}_2^T \bar{T}_2 = \begin{bmatrix} \alpha_1^2 + \beta_2^2 & \beta_2(\alpha_1 + \alpha_2) \\ \beta_2(\alpha_1 + \alpha_2) & \beta_2^2 + \alpha_2^2 + \beta_3^2 \end{bmatrix}, \quad T_2 = \begin{bmatrix} \alpha_1 & \beta_2 \\ \beta_2 & \alpha_2 \end{bmatrix}. \tag{3.52}
$$

We solve the equivalent problem $T_2^{-1}M^2 w = \xi w$ with $\alpha_1 = 0$ and find that

$$\xi_1^{(2)} = \frac{\alpha_2 - \sqrt{\alpha_2^2 + 4\beta_2^2 + 4\beta_3^2}}{2} = \frac{\alpha_2 - \sqrt{\alpha_2^2 + 4\beta_3^2 + O(\sin^2\varphi)}}{2}, \tag{3.53}$$

where β_2, α_2 and β_3 are given in (3.40) through (3.42). Note that this is a negative number bounded below independent of φ, and $\xi_1^{(m)}$ increases with m to approximate μ_1 from below. Therefore, in the first few MINRES iterations, $p_m(\mu_1) > 1$ if

$$\frac{\alpha_2 - \sqrt{\alpha_2^2 + 4\beta_2^2 + 4\beta_3^2}}{2} < -\left(\frac{1 - \sum_{k=2}^{m} \mu_1/\xi_k^{(m)}}{\sum_{k=2}^{m} 1/\xi_k^{(m)}}\right). \tag{3.54}$$

For some problems, $p_m(\mu_1) > 1$ holds in the initial MINRES iteration steps, but it will not take many iterations in practice before $p_m(\mu_1) < 1$ so that Theorem 3.3.2 can be applied and the bound in (3.10) becomes informative.

3.7 Concluding remarks

We have presented a detailed convergence analysis of three versions of MINRES to solve the linear systems in Rayleigh Quotient Iteration to find the lowest eigenpair of a symmetric positive definite matrix. Based on insight about the behavior of Ritz and harmonic Ritz values, our analysis includes qualitative and quantitative understanding of initial slow convergence of MINRES iterations, the main weakness of ordinary preconditioning without tuning in inexact RQI, the virtue of tuning, and the advantage of MINRES over SYMMLQ.

Using the idea of the BFGS formula in quasi-Newton methods, we propose a tuning method based on a rank-2 modification which guarantees positive definiteness of the symmetric tuned preconditioner. Other rank-2 modification formulas, such as DFP in quasi-Newton methods, could also be used.

Considering the performance of the three preconditioned MINRES solves on the last test problem, we speculate that our convergence analysis of MINRES on standard eigenvalue problems can be extended to generalized eigenvalue problems.

In fact, the inner solve of inexact RQI can be performed in two ways. The first way is to solve the shifted linear system $(A - \sigma I)y = x$ directly, and the second is to solve a simplified Jacobi-Davidson correction equation $(I - xx^*)(A - \sigma I)(I - xx^*)(dx) = -(A - \sigma I)x$ and let $y = x + dx$. The first methodology is studied in [75, 26] and this chapter; the second is investigated in [58]. It was shown in [27, 29] that for a given preconditioning matrix P, solving the shifted linear system by the full orthogonalization method (FOM) with $\mathbb{P}$ (the tuned version of P) is equivalent to solving the Jacobi-Davidson correction equation by FOM with P (untuned), in the sense that the inner iterates obtained by the two approaches in the same inner step are identical up to a scaling factor. This equivalence can be explained as follows. It can be shown that if the shifted linear system is preconditioned by $\mathbb{P}$, a good approximate solution can be computed in the first inner iteration. This approximate solution is also very close to the approximate desired eigenvector in the current outer iteration. Therefore, in the subsequent inner iteration steps, what is essentially done is to compute a correction to this approximate eigenvector. The corrected eigenvector obtained after this inner solve is expected to be a better approximation to the desired eigenvector. This effect is precisely consistent with what is achieved by the inner solves of the simplified Jacobi-Davidson method. We will discuss the connection between the motivation of tuning and that of the Jacobi-Davidson method in the next chapter for inexact subspace iteration.

4 Inexact subspace iteration

In this chapter, we study an inexact subspace iteration for solving generalized non-Hermitian eigenvalue problems with spectral transformation, with focus on a few strategies that help accelerate preconditioned iterative solution of the linear systems of equations arising in this context. We provide new insights into the preconditioner with tuning that has been studied for this algorithm applied to standard eigenvalue problems and propose a two-phase algorithm to use the tuned preconditioner in a simplified way to achieve similar performance for generalized problems. We discuss the connection between the two-phase algorithm and some methods for efficiently solving the linear systems arising in inexact inverse power method. In addition, we show that the cost of iterative solution of the linear systems can be further reduced by using deflation of converged Schur vectors, special starting vectors constructed from previously solved linear systems, and iterative linear solvers with subspace recycling.

4.1 Introduction

We have studied an inexact Rayleigh quotient iteration (RQI) for computing the lowest eigenpair of a Hermitian positive definite matrix in Chapter 3. RQI can also be applied to compute a simple eigenpair of a non-Hermitian matrix. Nevertheless, this method works with only a single vector and is generally not useful for computing several eigenpairs. In this situation, subspace iteration (also referred to as orthogonal or simultaneous iteration) can be used to compute a few dominant eigenpairs of a linear operator $\mathcal{A}$. As introduced in Chapter 2, subspace iteration is a straightforward block generalization of the power method which works with subspaces of a fixed dimension. The subspaces generated contain approximate desired Schur vectors that converge linearly as the (outer) iteration proceeds, if the matrix-vector products involving $\mathcal{A}$

are computed accurately.

In this chapter, we study an inexact subspace iteration for computing a few non-dominant eigenpairs of the generalized non-Hermitian eigenvalue problem $Av = \lambda Bv$. For this computation, $\mathcal{A}$ is chosen to be a proper shift-invert or Cayley transformation operator which maps the desired eigenvalues to well-separated dominant ones of a transformed problem. Without loss of generality, we use the notation $\mathcal{A} = A^{-1}B$, and we are interested in the k eigenvalues of $Av = \lambda Bv$ with smallest magnitude (i.e., k dominant eigenvalues of $\mathcal{A} = A^{-1}B$). This notation covers both shift-invert and Cayley transformation operators (2.10) with arbitrary shifts. For example, one can let $\widehat{A} = A - \sigma_1 B$ and $\widehat{B} = A - \sigma_2 B$, so that the generalized Cayley operator is $\mathcal{A} = \widehat{A}^{-1}\widehat{B}$. As a result, in each outer iteration, a linear system of the form $AY = BX$ with multiple right-hand sides need be solved. We assume that the matrices are so large that this linear solve has to be done by iterative solvers (inner iteration). The major concern of this chapter is to study several techniques to reduce the cost of the inner iteration.

Inexact subspace iteration for standard eigenvalue problems was studied in [65], where the linear convergence of outer iterations is established with the assumption that the inner solve is performed with reasonable accuracy. A second major contribution of [65] is the extension of the preconditioner with tuning [25, 26] to the iterative solution of the block systems $AY = BX$ arising in inexact subspace iteration. By the block-GMRES [68] convergence theory, it is shown there that tuning keeps the block-GMRES iteration counts roughly *constant* for solving these block linear systems, though the inner solve is required to be done with increasing accuracy as the outer iteration proceeds.

In this chapter, this idea is extended to generalized eigenvalue problems and is improved by a new two-phase algorithm. Specifically, we show that tuning can be limited to just one step of preconditioned block-GMRES to get an approximate solution, after which a correction equation can be solved to a *fixed* relative tolerance with proper preconditioned block linear solvers where tuning is *not* needed. We show that the effect of tuning is to reduce the residual in a special way, and that this effect can be also achieved by other means, in particular by solving a small least squares problem. Moreover, we show that the two-phase strategy is closely related to an inverse correction scheme presented in [67, 33] and the residual inverse power method in [87].

The second phase of this algorithm, in addition to using a simplified preconditioning strategy, can also be simplified in other ways to achieve additional reduction of inner iteration cost. We explore three techniques to attain the extra speedup:

1. *Deflation of converged Schur vectors* (see [86]) – Once some Schur vectors have converged, they are deflated from the block linear systems in subsequent outer iterations, so that the block size becomes smaller. This approach is independent of the way the block linear systems are solved.

2. *Special starting vector* – We find that the right hand sides of a few successive correction equations are often close to being linearly dependent; therefore an appropriate linear combination of the solutions to previously solved correction equations can be used as a good starting vector for solving the current one.

3. *Subspace Recycling* – Linear solvers with recycled subspaces (see [61]) can be used to solve the sequence of correction equations, so that the search space for each solve does not need to be built from scratch. In addition, if the same preconditioner is used for all correction equations, the recycled subspaces available from solving one equation can be used directly for the next without being transformed by additional preconditioned matrix-vector products.

We discuss the effectiveness of these ideas and show by numerical experiments that they generally result in significant savings in the number of preconditioned matrix-vector products performed in inner iterations.

An outline of this chapter is as follows. In Section 2, we describe the inexact subspace iteration for generalized non-Hermitian eigenvalue problems, restate some preliminary results taken from [65] about block decomposition of matrices, and discuss a new tool for measuring closeness of two subspaces. In Section 3, we briefly discuss the behavior of unpreconditioned and preconditioned block-GMRES without tuning for solving the block linear systems arising in inexact subspace iteration, and present new insights into tuning that lead to our two-phase strategy to solve the block linear systems. In Section 4, we discuss deflation of converged Schur vectors, special starting vector and linear solvers with recycled subspaces and the effectiveness of the combined use of these techniques for solving the block systems. Section 5 includes a series of numerical experiments to show the performance of our algorithm for problems from Matrix Market [51] and those arising from linear stability analysis of models of two-dimensional incompressible flows. We finally make concluding remarks in Section 6.

4.2 Inexact Subspace Iteration and Preliminary Results

In this section, we review an inexact subspace iteration for the generalized non-Hermitian eigenvalue problem $Av = \lambda Bv$ ($A, B \in \mathbb{C}^{n\times n}$) with spectral transformation, block Schur and eigen-decomposition of matrices, and metrics that measure the error of the current approximate invariant subspace.

4.2.1 Inexact Subspace Iteration

Algorithm 4.1 describes the inexact subspace iteration.

In Step 1, $\mathcal{X}^{(i)}$ is the space spanned by the current outer iterate $X^{(i)}$. The error of $X^{(i)}$ is defined as the sine of the largest principal angle between $A\mathcal{X}^{(i)}$ and $B\mathcal{X}^{(i)}$. It decreases to zero as $X^{(i)}$ converges to an invariant subspace of the matrix pencil (A, B). This error can be computed by MATLAB's function `subspace` based on

Algorithm 4.1 Inexact Subspace Iteration with $\mathcal{A} = A^{-1}B$

```
Given $\delta \geq 0$ and $X^{(0)} \in \mathbb{C}^{n\times p}$ with $X^{(0)*}X^{(0)} = I \quad (k \leq p)$
for $i = 0, 1, ...,$ until $k$ Schur vectors converge do
  1. Compute the error $e^{(i)} = \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})$
  2. Solve $AY^{(i)} = BX^{(i)}$ inexactly such that
  the relative residual norm$\frac{\|BX^{(i)}-AY^{(i)}\|}{\|BX^{(i)}\|} \leq \delta e^{(i)}$
  3. Perform the Schur-Rayleigh-Ritz procedure
  to get $X^{(i+1)}$ with orthonormal columns from $Y^{(i)}$
  and test for convergence
end for
```

singular value decomposition (see Algorithm 12.4.3 of [32]), and will be discussed in detail in Proposition 4.2.2.

The Schur-Rayleigh-Ritz (SRR) procedure (see Chapter 6.1 of [86]) in Step 3 will be applied to deflate converged Schur vectors. The procedure and its use for deflation will be explained in detail in Section 4.4. The most computationally expensive part of the algorithm is Step 2, which requires an inexact solve of the block linear system $AY^{(i)} = BX^{(i)}$. The major concern of this chapter is to reduce the cost of this solve.

4.2.2 Block eigen-decomposition

To briefly review the basic notations and description of the generalized eigenvalue problem, we restate some results from [65] on block eigen-decomposition of matrices and study a new tool to measure the error of $X^{(i)}$ for generalized problems. To simplify our exposition, we assume that B is nonsingular. This assumption is valid for problems arising in a variety of applications. In addition, though B is only positive semi-definite in linear stability analysis of incompressible flows, one can instead solve some related eigenvalue problems with nonsingular B that share the same finite eigenvalues as the original problem; see [11] for details.

As B is nonsingular, we assume that the eigenvalues of $B^{-1}A$ are ordered so that

$$0 < |\lambda_1| \leq |\lambda_2| \leq ... \leq |\lambda_p| < |\lambda_{p+1}| \leq ... \leq |\lambda_n|.$$

The Schur decomposition of $B^{-1}A$ can be written in block form as

$$B^{-1}A = \begin{bmatrix} V_1, \ V_1^{\perp} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} V_1, \ V_1^{\perp} \end{bmatrix}^*, \tag{4.1}$$

where $\begin{bmatrix} V_1, \ V_1^{\perp} \end{bmatrix}$ is a unitary matrix with $V_1 \in \mathbb{C}^{n\times p}$ and $V_1^{\perp} \in \mathbb{C}^{n\times(n-p)}$, $T_{11} \in \mathbb{C}^{p\times p}$ and $T_{22} \in \mathbb{C}^{(n-p)\times(n-p)}$ are upper triangular, $\lambda(T_{11}) = \{\lambda_1, ..., \lambda_p\}$ and $\lambda(T_{22}) = \{\lambda_{p+1}, ..., \lambda_n\}$. Since T_{11} and T_{22} have disjoint spectra, there is a unique solution $Q \in \mathbb{C}^{p\times(n-p)}$ to the Sylvester equation $QT_{22} - T_{11}Q = T_{12}$ (see Section 1.5, Chapter 1 of [86]). Then $B^{-1}A$ can be transformed to block-diagonal form as follows:

$$\begin{aligned} B^{-1}A &= \begin{bmatrix} V_1, \ V_1^{\perp} \end{bmatrix} \begin{bmatrix} I & Q \\ 0 & I \end{bmatrix} \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix} \begin{bmatrix} V_1, \ V_1^{\perp} \end{bmatrix}^* \\ &= \begin{bmatrix} V_1, \ (V_1Q + V_1^{\perp}) \end{bmatrix} \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} (V_1 - V_1^{\perp}Q^*), \ V_1^{\perp} \end{bmatrix}^* \\ &= \begin{bmatrix} V_1, \ (V_1Q + V_1^{\perp})Q_D^{-1} \end{bmatrix} \begin{bmatrix} T_{11} & 0 \\ 0 & Q_DT_{22}Q_D^{-1} \end{bmatrix} \begin{bmatrix} (V_1 - V_1^{\perp}Q^*), \ V_1^{\perp}Q_D \end{bmatrix}^* \\ &= [V_1, \ V_2] \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} [W_1, \ W_2]^* = [V_1, \ V_2] \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} [V_1, \ V_2]^{-1} \end{aligned} \tag{4.2}$$

where $Q_D = (I + Q^*Q)^{1/2}$, $V_2 = (V_1Q + V_1^{\perp})Q_D^{-1}$ with orthonormal columns, $K = T_{11}$, $M = Q_DT_{22}Q_D^{-1}$ with the same spectrum as T_{22}, $W_1 = V_1 - V_1^{\perp}Q^*$ and $W_2 = V_1^{\perp}Q_D$ such that $\begin{bmatrix} W_1, \ W_2 \end{bmatrix}^* = \begin{bmatrix} V_1, \ V_2 \end{bmatrix}^{-1}$. From the last expression of (4.2), we have

$$AV_1 = BV_1K, \quad \text{and} \quad AV_2 = BV_2M. \tag{4.3}$$

Recall that we want to compute V_1 and corresponding eigenvalues (the spectrum of K) by inexact subspace iteration.

4.2.3 Tools to measure the error

The basic tool to measure the deviation of $\mathcal{X}^{(i)} = \text{span}\{X^{(i)}\}$ from $\mathcal{V}_1 = \text{span}\{V_1\}$ is the sine of the largest principal angle between $\mathcal{X}^{(i)}$ and $\mathcal{V}_1$ defined as (see [65] and references therein)

$$\begin{aligned}(4.4) \qquad \sin\angle(\mathcal{X}^{(i)}, \mathcal{V}_1) &= \|(V_1^{\perp})^* X^{(i)}\| = \|X^{(i)}(X^{(i)})^* - V_1V_1^*\| \\ &= \min_{Z\in\mathbb{C}^{p\times p}} \|X^{(i)} - V_1Z\| = \min_{Z\in\mathbb{C}^{p\times p}} \|V_1 - X^{(i)}Z\|.\end{aligned}$$

This definition depends on the fact that both $X^{(i)}$ and V_1 have orthonormal columns.

We assume that $X^{(i)}$ has the following decomposition

$$(4.5) \qquad X^{(i)} = V_1C^{(i)} + V_2S^{(i)},$$

where $C^{(i)} = W_1^*X^{(i)} \in \mathbb{C}^{p\times p}$, $S^{(i)} = W_2^*X^{(i)} \in \mathbb{C}^{(n-p)\times p}$. Intuitively, $\|S^{(i)}\| \to 0$ as $\mathcal{X}^{(i)} \to \mathcal{V}_1$. Properties of $C^{(i)}$ and the equivalence of several metrics are given in the following proposition.

Proposition 4.2.1 (PROPOSITION 2.1 in [65]) *Suppose $X^{(i)}$ is decomposed as in (4.5). Let $s^{(i)} = \|S^{(i)}(C^{(i)})^{-1}\|$ and $t^{(i)} = s^{(i)}\|C^{(i)}\|$. Then*

1) $C^{(i)}$ is nonsingular and thus $s^{(i)}$ is well-defined. The singular values of $C^{(i)}$ satisfy

$$(4.6) \qquad 0 < 1 - \|S^{(i)}\| \le \sigma_k(C^{(i)}) \le 1 + \|S^{(i)}\|, \; k = 1, 2, ..., p$$

and $C^{(i)} = U^{(i)} + \Upsilon^{(i)}$, where $U^{(i)}$ is unitary and $\|\Upsilon^{(i)}\| \le \|S^{(i)}\| < 1$.

2a) $\sin\angle(\mathcal{X}^{(i)}, \mathcal{V}_1) \le \|S^{(i)}\| \le s^{(i)} \le \left(\frac{1+\|S^{(i)}\|}{1-\|S^{(i)}\|}\right)\|S^{(i)}\|$

2b) $\sin\angle(\mathcal{X}^{(i)}, \mathcal{V}_1) \le t^{(i)} \le \frac{\|S^{(i)}\|}{1-\|S^{(i)}\|}$

2c) $\|S^{(i)}\| \le \sqrt{1+\|Q\|^2}\sin\angle(\mathcal{X}^{(i)}, \mathcal{V}_1)$.

The proposition states that as $\mathcal{X}^{(i)} \to \mathcal{V}_1$, $C^{(i)}$ gradually approximates a unitary matrix, and $\sin\angle(\mathcal{X}^{(i)}, \mathcal{V}_1)$, $\|S^{(i)}\|$, $s^{(i)}$ and $t^{(i)}$ are essentially equivalent measures of the error. These quantities are not computable since $\mathcal{V}_1$ is not available. However, the computable quantity $\sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})$ in Step 1 of Algorithm 4.1 is equivalent to $\|S^{(i)}\|$, as the following proposition shows.

Proposition 4.2.2 *Let $X^{(i)}$ be decomposed as in (4.5). Then*

$$c_1\|S^{(i)}\| \leq \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}) \leq c_2\|S^{(i)}\|, \tag{4.7}$$

where c_1 and c_2 are constants independent of the progress of subspace iteration.

Proof We first show that as $\mathcal{X}^{(i)} \to \mathcal{V}_1$, $A\mathcal{X}^{(i)} \approx B\mathcal{X}^{(i)}$. In fact, from (4.3) and (4.5) we have

$$\begin{aligned} BX^{(i)} = BV_1C^{(i)} + BV_2S^{(i)} &= AV_1K^{-1}C^{(i)} + AV_2M^{-1}S^{(i)} \\ &= A(X^{(i)} - V_2S^{(i)})(C^{(i)})^{-1}K^{-1}C^{(i)} + AV_2M^{-1}S^{(i)} \\ &= AX^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)} - AV_2\left(S^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)} - M^{-1}S^{(i)}\right). \end{aligned} \tag{4.8}$$

Roughly speaking, $AX^{(i)}$ and $BX^{(i)}$ can be transformed to each other by postmultiplying $(C^{(i)})^{-1}K^{-1}C^{(i)}$ or its inverse, with a small error proportional to $\|S^{(i)}\|$.

Let $D_A^{(i)} = (X^{(i)*}A^*AX^{(i)})^{-1/2}$, $D_B^{(i)} = (X^{(i)*}B^*BX^{(i)})^{-1/2} \in \mathbb{C}^{p\times p}$, so that both $AX^{(i)}D_A^{(i)}$ and $BX^{(i)}D_B^{(i)}$ have orthonormal columns. Then by (4.4)

$$\begin{aligned} \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}) &= \min_{Z\in\mathbb{C}^{p\times p}} \|AX^{(i)}D_A^{(i)} - BX^{(i)}D_B^{(i)}Z\| \\ &\leq \left\|AX^{(i)}D_A^{(i)} - BX^{(i)}D_B^{(i)}\left((D_B^{(i)})^{-1}(C^{(i)})^{-1}KC^{(i)}D_A^{(i)}\right)\right\| \\ &= \left\|\left(AX^{(i)} - BX^{(i)}(C^{(i)})^{-1}KC^{(i)}\right)D_A^{(i)}\right\| \\ &= \left\|AV_2\left(S^{(i)} - M^{-1}S^{(i)}(C^{(i)})^{-1}KC^{(i)}\right)D_A^{(i)}\right\| \qquad \text{(see (4.8))} \\ &= \left\|BV_2\left(MS^{(i)} - S^{(i)}(C^{(i)})^{-1}KC^{(i)}\right)D_A^{(i)}\right\| \qquad \text{(see (4.3))} \\ &\leq \|BV_2\|\|D_A^{(i)}\|\|\mathcal{S}_i\|\|S^{(i)}\|, \end{aligned} \tag{4.9}$$

where $\mathcal{S}_i$ is the Sylvester operator $G \to \mathcal{S}_i(G) : MG - G(C^{(i)})^{-1}KC^{(i)}$. Note that as i

increases, $\|\mathcal{S}_i\| \to \|\mathcal{S}\|$ where $\mathcal{S} : G \to \mathcal{S}(G) = MG - GK$, as the following derivation shows:

$$
\begin{aligned}
(4.10) \qquad \|\mathcal{S}_i\| &= \sup_G \frac{\|MG - G(C^{(i)})^{-1}KC^{(i)}\|}{\|G\|} \qquad (G \in \mathbb{C}^{(n-p)\times p}) \\
&= \sup_G \frac{\left\|\Big(M\big(G(C^{(i)})^{-1}\big) - \big(G(C^{(i)})^{-1}\big)K\Big)C^{(i)}\right\|}{\|\,(G(C^{(i)})^{-1})\,C^{(i)}\|},
\end{aligned}
$$

and therefore

$$
\begin{aligned}
(4.11) \qquad & \sup_{\tilde{G}} \frac{\|M\tilde{G} - \tilde{G}K\|}{\|\tilde{G}\|\kappa(C^{(i)})} \le \|\mathcal{S}_i\| \le \sup_{\tilde{G}} \frac{\|M\tilde{G} - \tilde{G}K\|\kappa(C^{(i)})}{\|\tilde{G}\|} \qquad (\tilde{G} \in \mathbb{C}^{(n-p)\times p}) \\
\text{or} \qquad & \|S\|/\kappa(C^{(i)}) \le \|\mathcal{S}_i\| \le \|S\|\kappa(C^{(i)}).
\end{aligned}
$$

As $1 \le \kappa(C^{(i)}) = \frac{\sigma_{max}(C^{(i)})}{\sigma_{min}(C^{(i)})} \le \frac{1+\|S^{(i)}\|}{1-\|S^{(i)}\|}$ (Proposition 4.2.1, 1)) and $\|S^{(i)}\| \to 0$, $\|\mathcal{S}_i\| \to \|\mathcal{S}\|$ follows. In addition, as $D_A^{(i)}$ is derived from the projection of A^*A onto $\mathcal{X}^{(i)} \to \mathcal{V}_1$, $\|D_A^{(i)}\|$ is bounded above by some constant independent of i. The upper bound in (4.7) is thus established.

To study the lower bound, we have

$$
\begin{aligned}
(4.12) \qquad \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}) &= \min_{Z\in\mathbb{C}^{p\times p}} \|AX^{(i)}D_A^{(i)} - BX^{(i)}D_B^{(i)}Z\| \\
&= \min_{Z\in\mathbb{C}^{p\times p}} \left\|B\big(B^{-1}AX^{(i)} - X^{(i)}D_B^{(i)}Z(D_A^{(i)})^{-1}\big)D_A^{(i)}\right\| \\
&\ge \min_{\tilde{Z}\in\mathbb{C}^{p\times p}} \|B^{-1}AX^{(i)} - X^{(i)}\tilde{Z}\|\sigma_{min}(B)\sigma_{min}(D_A^{(i)}).
\end{aligned}
$$

Let $\sigma^{(i)} = \sigma_{min}(B)\sigma_{min}(D_A^{(i)}) = \sigma_{min}(B)\|(X^{(i)*}A^*AX^{(i)})^{1/2}\|$. Since the minimizer $\tilde{Z}$ in the last inequality is $K^{(i)} = X^{(i)*}B^{-1}AX^{(i)}$ (see Chapter 4, Theorem 2.6 of [86]), we have

$$
\begin{aligned}
(4.13)\qquad & \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}) \geq \sigma^{(i)}\|B^{-1}AX^{(i)} - X^{(i)}K^{(i)}\| \\
\geq\ & \sigma^{(i)}\|(V_1^{\perp})^*(B^{-1}AX^{(i)} - X^{(i)}K^{(i)})\| \qquad (\,\|V_1^{\perp}\| = 1\,) \\
=\ & \sigma^{(i)}\|T_{22}(V_1^{\perp})^*X^{(i)} - (V_1^{\perp})^*X^{(i)}K^{(i)}\| \quad (\,(V_1^{\perp})^*B^{-1}A = T_{22}(V_1^{\perp})^*;\ \text{see (4.1)}\,) \\
\geq\ & \sigma^{(i)}\mathrm{sep}(T_{22}, K^{(i)})\|(V_1^{\perp})^*X^{(i)}\| = \sigma^{(i)}\mathrm{sep}(T_{22}, K^{(i)})\sin\angle(\mathcal{V}_1, \mathcal{X}^{(i)}) \\
\geq\ & \sigma^{(i)}\mathrm{sep}(T_{22}, K^{(i)})(1 + \|Q\|^2)^{-1/2}\|S^{(i)}\|. \qquad (\text{Proposition 4.2.1, 2c}))
\end{aligned}
$$

For similar reasons for $\|D_A^{(i)}\|$, $\sigma^{(i)} = \sigma_{min}(B)\sigma_{min}(D_A^{(i)})$ is bounded below by some constant independent of i. Moreover, as $\mathcal{X}^{(i)} \to \mathcal{V}_1$, $K^{(i)} = X^{(i)*}B^{-1}AX^{(i)} \to K$ up to a unitary transformation, and hence $\mathrm{sep}(T_{22}, K^{(i)}) \to \mathrm{sep}(T_{22}, K)$ (see Chapter 4, Theorem 2.11 in [86]). This concludes the proof. ∎

Remark 2.1. In [65] the authors use $\|AX^{(i)} - X^{(i)}\left(X^{(i)*}AX^{(i)}\right)\|$ to estimate the error of $X^{(i)}$ for standard eigenvalue problems, and show that this quantity is essentially equivalent to $\|S^{(i)}\|$. The $p \times p$ matrix $X^{(i)*}AX^{(i)}$ is called the (block) Rayleigh quotient of A (see Chapter 4, Definition 2.7 of [86]). For the generalized eigenvalue problem, we have not seen an analogous quantity in the literature. However, Proposition 4.2.2 shows that $\sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})$ is a convenient error estimate.

4.3 Convergence Analysis of Inexact Subspace Iteration

We first demonstrate the linear convergence of the outer iteration of Algorithm 4.1, which follows from Theorem 3.1 in [65]. In fact, replacing A^{-1} by $A^{-1}B$ in the proof therein, we have

$$(4.14)\qquad t^{(i+1)} \leq \|K\|\|M^{-1}\|\frac{t^{(i)} + \|W_2^*B^{-1}\|\,\|(C^{(i)})^{-1}\|\,\|R^{(i)}\|}{1 - \|W_1^*B^{-1}\|\,\|(C^{(i)})^{-1}\|\,\|R^{(i)}\|},$$

where $\|(C^{(i)})^{-1}\| = 1/\sigma_{min}(C^{(i)}) \to 1$, and

$$\begin{aligned}(4.15) \qquad \|R^{(i)}\| &= \|BX^{(i)} - AY^{(i)}\| \leq \delta\|BX^{(i)}\| \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}) \\ &\leq \delta C_2\|BX^{(i)}\|\sqrt{1+\|Q\|^2}t^{(i)}. \qquad \text{(see Proposition 4.2.1)}\end{aligned}$$

Thus $\mathcal{X}^{(i)} \to \mathcal{V}_1$ linearly for $\|K\|\|M^{-1}\| < 1$ and small enough δ.

In this section, we investigate the convergence of unpreconditioned and preconditioned block-GMRES without tuning for solving $AY^{(i)} = BX^{(i)}$ to the prescribed tolerance, and provide new perspectives on tuning that lead to a new two-phase strategy for solving this block system.

4.3.1 Unpreconditioned and preconditioned block-GMRES with no tuning

It is shown in [65] that when solving the block linear systems $AY^{(i)} = X^{(i)}$ arising in inexact subspace iteration for standard eigenvalue problems, as the tolerance of this solve decreases with the outer iteration progress, the unpreconditioned block-GMRES iteration counts remain roughly constant, but the number of preconditioned block-GMRES iterations without tuning increases. The reason is that the right hand side $X^{(i)}$ is an approximate invariant subspace of the system matrix A, whereas with preconditioning, there is no reason for $X^{(i)}$ to bear such a relation to the preconditioned system matrix AP^{-1} (assuming right preconditioning is used).

For generalized eigenvalue problems, however, both the unpreconditioned and preconditioned block-GMRES iteration counts increase progressively. To see this point, we study the block spectral decomposition of the system matrix and right hand side and review the block-GMRES convergence theory. We present the analysis of the unpreconditioned solve; the results apply verbatim to the preconditioned solve.

We first review a generic convergence result of block-GMRES given in [65]. Let G be a matrix of order n where the p smallest eigenvalues are separated from its other

$n-p$ eigenvalues. As in (4.2), we can block diagonalize G as

$$G = \begin{bmatrix} V_{G1}, \ V_{G2} \end{bmatrix} \begin{bmatrix} K_G & 0 \\ 0 & M_G \end{bmatrix} \begin{bmatrix} V_{G1}, \ V_{G2} \end{bmatrix}^{-1}, \tag{4.16}$$

where $V_{G1} \in \mathbb{C}^{n \times p}$ and $V_{G2} \in \mathbb{C}^{n \times (n-p)}$ have orthonormal columns, $\lambda(K_G)$ are the p smallest eigenvalues of G, and $\lambda(M_G)$ are the other eigenvalues of G. Recall the definitions of the numerical range $W(M_G) = \left\{ \frac{z^* M_G z}{z^* z} : z \in \mathbb{C}^{n-p}, z \neq 0 \right\}$ and the ϵ-pseudospectrum $\lambda_\epsilon(M_G) = \{\lambda \in \mathbb{C} : \sigma_{min}(\lambda I - M_G) \leq \epsilon\}$. The role of the right hand side in the convergence of block-GMRES is described in the following lemma, which follows immediately from Theorem 3.7 of [65].

Lemma 4.3.1 *Assume the numerical range $W(M_G)$ or the ϵ-pseudospectrum $\lambda_\epsilon(M_G)$ is contained in a convex closed bounded set E in the complex plane with $0 \notin E$. Suppose block-GMRES is used to solve $GY = Z$ where $Z \in \mathbb{C}^{n \times p}$ can be decomposed as $Z = V_{G1} C_G + V_{G2} S_G$ with V_{G1} and V_{G2} given in (4.16). Here $S_G \in \mathbb{C}^{(n-p) \times p}$, and we assume $C_G \in \mathbb{C}^{p \times p}$ is nonsingular. Let Y_k be the approximate solution of $GY = Z$ obtained in the k-th block-GMRES iteration with $Y_0 = 0$. If*

$$k \geq 1 + C_a \left(C_b + \log \frac{\|C_G\| \|S_G C_G^{-1}\|}{\|Z\| \tau} \right), \tag{4.17}$$

then $\frac{\|Z - GY_k\|}{\|Z\|} \leq \tau$. Here C_a and C_b are constants that depend on the spectrum of G.

Remark 3.1. For details about C_a and C_b, see [65], [40] and [21]. These details have minimal impact on our subsequent analysis.

Remark 3.2. This generic convergence result can be applied to any specific block linear systems with or without preconditioning. For example, to study the behavior of unpreconditioned block-GMRES for solving $AY^{(i)} = BX^{(i)}$, let $G = A$ in (4.16) and Lemma 4.3.1, and decompose relevant sets of column vectors in terms of V_{A1} and V_{A2}.

In the following, we will assume that for nontrivial B ($B \neq I$), there is no pathological or trivial connection between the decomposition of (4.16) for the case $G = A$ and that of $B^{-1}A$ in (4.2). Specifically, we assume there exist a *nonsingular* $C_1 \in \mathbb{C}^{p\times p}$ and a *full rank* $S_1 \in \mathbb{C}^{(n-p)\times p}$ for which both $\|C_1\|$ and $\|S_1C_1^{-1}\|$ are not too small or too large, such that

$$V_1 = V_{A1}C_1 + V_{A2}S_1. \tag{4.18}$$

That is, V_1 of the generalized eigenvalue problem is not dominated by either invariant subspace of A specified in (4.16) (with $G = A$). This assumption is generally far from stringent, and is consistent with our numerical experiences. Similarly, let the decomposition of V_2 be $V_2 = V_{A1}C_2 + V_{A2}S_2$ with $C_2 \in \mathbb{C}^{p\times(n-p)}$ and $S_2 \in \mathbb{C}^{(n-p)\times(n-p)}$.

Lemma 4.3.1 and the assumptions above lead to the following theorem, which qualitatively describes the behavior of block-GMRES for solving $AY^{(i)} = BX^{(i)}$.

Theorem 4.3.2 *Assume that unpreconditioned block-GMRES is used to solve the linear system $AY^{(i)} = BX^{(i)}$ to the prescribed tolerance in Step 2 of Algorithm 4.1. If the assumption (4.18) holds, and $\mathcal{X}^{(i)} \to \mathcal{V}_1$ linearly, then the bound on block-GMRES iteration counts that guarantees satisfaction of the prescribed the tolerance increases as the outer iteration proceeds.*

Proof With (4.3), (4.5), (4.16) and (4.18), we have

$$\begin{aligned} BX^{(i)} &= BV_1C^{(i)} + BV_2S^{(i)} = AV_1K^{-1}C^{(i)} + AV_2M^{-1}S^{(i)} \\ &= A(V_{A1}C_1 + V_{A2}S_1)K^{-1}C^{(i)} + A(V_{A1}C_2 + V_{A2}S_2)M^{-1}S^{(i)} \\ &= (V_{A1}K_AC_1 + V_{A2}M_AS_1)K^{-1}C^{(i)} + (V_{A1}K_AC_2 + V_{A2}M_AS_2)M^{-1}S^{(i)} \\ &= V_{A1}K_A(C_1K^{-1}C^{(i)} + C_2M^{-1}S^{(i)}) + V_{A2}M_A(S_1K^{-1}C^{(i)} + S_2M^{-1}S^{(i)}) \\ &= V_{A1}C_A^{(i)} + V_{A2}S_A^{(i)}. \end{aligned} \tag{4.19}$$

Since $\|S^{(i)}\| \to 0$ and $\sigma_k(C^{(i)}) \to 1 (k = 1, 2, ..., p)$ (see Proposition 4.2.1, 1)), we have $\|C_A^{(i)}\| \to \|K_A C_1 K^{-1}\|$ and $\|S_A^{(i)}(C_A^{(i)})^{-1}\| \to \|M_A S_1 C_1^{-1} K_A^{-1}\|$, both of which are moderate quantities under our assumption for (4.18).

From Step 2 of Algorithm 4.1 and (4.9), the relative tolerance for $AY^{(i)} = BX^{(i)}$ is $\tau = \delta \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}) \le \delta \|BV_2\| \|D_A^{(i)}\| \|\mathcal{S}_i\| \|S^{(i)}\| \to 0$. Then from (4.17) and (4.19), the lower bound on the unpreconditioned block-GMRES iteration counts which guarantees satisfaction of the prescribed tolerance is

$$k^{(i)} \ge 1 + C_a \left(C_b + \log \frac{\|C_A^{(i)}\| \|S_A^{(i)}(C_A^{(i)})^{-1}\|}{\delta \|BX^{(i)}\| \|BV_2\| \|D_A^{(i)}\| \|\mathcal{S}_i\| \|S^{(i)}\|} \right). \tag{4.20}$$

Note that $\|BX^{(i)}\| \to \|BV_1\|$, $\|D_A^{(i)}\| \to \|(V_1 A^* A V_1)^{-1/2}\|$ and $\|\mathcal{S}_i\| \to \|\mathcal{S}\|$ (see the proof of Proposition 4.2.2). Therefore all terms in the argument of the logarithm operator approach some nonzero limit except $\|S^{(i)}\| \to 0$, and hence the bound on $k^{(i)}$ which guarantees satisfaction of the tolerance increases as the outer iteration proceeds. ▮

Remark 3.3. This result only shows that an *upper bound* on $k^{(i)}$ increases but doesn't establish that the actual block-GMRES iteration counts will grow. However, numerical experiments in [65] and Section 4.5 this chapter do indicate that this result is indicative of performance. The gradual increase of inner iteration counts depends on the assumption of (4.18) that V_1 has "regular" components of V_{A1} and V_{A2}. This assumption guarantees that $B\mathcal{X}^{(i)}$ does not approximate the invariant subspace $\mathcal{V}_{A1}$ of A. The proof also applies word for word to the preconditioned solve without tuning: one only needs to replace (4.16) by the decomposition of the preconditioned system matrix AP^{-1} and write $BX^{(i)}$ in terms of the invariant subspace of AP^{-1}.

4.3.2 Preconditioned block-GMRES with tuning

To accelerate the iterative solution of the block linear system arising in inexact subspace iteration, [65] proposes and analyzes a new type of preconditioner with tuning.

Tuning constructs a special low-rank update of the existing preconditioner, so that the right hand side of the preconditioned system is an approximate eigenvector or invariant subspace of the preconditioned system matrix with tuning. Specifically, given a preconditioner P, the tuned preconditioner in the i-th outer iteration can be defined as

$$\mathbb{P}^{(i)} = P + (AX^{(i)} - PX^{(i)})X^{(i)*}, \tag{4.21}$$

from which follow $\mathbb{P}^{(i)}X^{(i)} = AX^{(i)}$ and $A(\mathbb{P}^{(i)})^{-1}(AX^{(i)}) = AX^{(i)}$. In other words, $A\mathcal{X}^{(i)}$ is an invariant subspace of $A(\mathbb{P}^{(i)})^{-1}$ with eigenvalue 1. Intuitively, as $\mathcal{X}^{(i)} \to \mathcal{V}_1$, we have $\mathcal{X}^{(i)} \approx A\mathcal{X}^{(i)}$ for the standard eigenvalue problem, or $B\mathcal{X}^{(i)} \approx A\mathcal{X}^{(i)}$ for the generalized problem. Therefore, the right hand side of $A(\mathbb{P}^{(i)})^{-1}\tilde{Y}^{(i)} = X^{(i)}$ or $A(\mathbb{P}^{(i)})^{-1}\tilde{Y}^{(i)} = BX^{(i)}$ (with $Y^{(i)} = (\mathbb{P}^{(i)})^{-1}\tilde{Y}^{(i)}$) spans an approximate invariant subspace of $A(\mathbb{P}^{(i)})^{-1}$. The difficulty of block-GMRES without tuning discussed in subsection 4.3.1 is thus resolved, and the block-GMRES iteration counts with tuning do not increase with the progress of the outer iteration (see Theorem 4.5 of [65]).

The matrix-vector product involving $(\mathbb{P}^{(i)})^{-1}$ is built from that for P^{-1} using the Sherman-Morrison-Woodbury formula as follows:

$$(\mathbb{P}^{(i)})^{-1} = \left(I - (P^{-1}AX^{(i)} - X^{(i)})\big(X^{(i)*}P^{-1}AX^{(i)}\big)^{-1}X^{(i)*}\right)P^{-1}. \tag{4.22}$$

Note that $P^{-1}AX^{(i)} - X^{(i)}$ and $X^{(i)*}P^{-1}AX^{(i)}$ can be computed before the block-GMRES iteration. In each block-GMRES step, $(\mathbb{P}^{(i)})^{-1}$ requires an additional p^2 inner products of vectors of length n, a dense matrix-matrix division of size $p \times p$, and a multiplication of a dense matrix of size $n \times p$ with a $p \times p$ matrix. This extra cost is relatively small in general, but it is not free.

We now provide a new two-phase algorithm for solving $AY^{(i)} = BX^{(i)}$, which essentially eliminates the overhead of tuning but keeps the block-GMRES iteration counts from progressively increasing. The strategy provides some new perspectives on the use of tuning. In addition, we will discuss some connections between this algorithm and the methods in [67, 33] and [87].

Algorithm 4.2 Two-phase strategy for solving $AY^{(i)} = BX^{(i)}$

```
1. Apply a single step of preconditioned block-GMRES with tuning to get an approximate solution $Y_1^{(i)}$
2. Solve the correction equation $A\,dY^{(i)} = BX^{(i)} - AY_1^{(i)}$ with proper preconditioned iterative solver to get an approximate solution $dY_k^{(i)}$, so that $Y_{k+1}^{(i)} = Y_1^{(i)} + dY_k^{(i)}$ satisfies $\frac{\|BX^{(i)}-AY_{k+1}^{(i)}\|}{\|BX^{(i)}\|} \leq \delta \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})$, or equivalently, $dY_k^{(i)}$ satisfies $\frac{\|(BX^{(i)}-AY_1^{(i)})-A\,dY_k^{(i)}\|}{\|BX^{(i)}-AY_1^{(i)}\|} \leq \frac{\delta \sin\angle(A\mathcal{X}^{(i)},B\mathcal{X}^{(i)})\|BX^{(i)}\|}{\|BX^{(i)}-AY_1^{(i)}\|}$
```

Note in particular that tuning need *not* be used to solve the correction equation, and thus we can work with a fixed preconditioned system matrix for the correction equation in all outer iterations.

Obviously, Phase II can be equivalently stated as follows: solve $AY^{(i)} = BX^{(i)}$ with proper preconditioned iterative solver and starting vector $Y_1^{(i)}$ from Phase I. The phrasing in Algorithm 4.2 is intended to illuminate the connection between this strategy and the methods in [67, 33] and [87].

The analysis of Algorithm 4.2 is given in the following main theorem. For this, we make an assumption concerning the right hand side of the correction equation analogous to the assumption made for (4.18): with

$$BX^{(i)} - AY_1^{(i)} = V_{A1}C_{ceq}^{(i)} + V_{A2}S_{ceq}^{(i)} \tag{4.23}$$

where V_{A1} and V_{A2} have orthonormal columns, we assume that $\|S_{ceq}^{(i)}(C_{ceq}^{-1})^{(i)}\| = O(1)$. Given the fact that both V_{A1} and V_{A2} have orthonormal columns, we further assume that $\|C_{ceq}\|$ is proportional to $\|BX^{(i)} - AY_1^{(i)}\|$. This implies that the term $\frac{\|C_{ceq}^{(i)}\|\|S_{ceq}^{(i)}(C_{ceq}^{-1})^{(i)}\|}{\|BX^{(i)}-AY_1^{(i)}\|}$, which appears in (4.17), does not depend on i. We have no proof of these assumptions but they are consistent with all our numerical experience. In the subsequent derivation, let $e_j \in \mathbb{R}^p$ be a standard unit basis vector with 1 in entry

j and zero in other entries.

Theorem 4.3.3 *Suppose the two-phase strategy (Algorithm 4.2) is used to solve* $AY^{(i)} = BX^{(i)}$*. Then* $Y_1^{(i)} = X^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)}F + \Delta^{(i)}$ *where* $F = [f_1, \dots, f_p]$ *with* $f_j = \operatorname{argmin}_{f\in\mathbb{C}^p} \|BX^{(i)}e_j - A(\mathbb{P}^{(i)})^{-1}BX^{(i)}f\|$ *and* $\|\Delta^{(i)}\| = O(\|S^{(i)}\|)$*. In addition, the residual norm* $\|BX^{(i)} - AY_1^{(i)}\| = O(\|S^{(i)}\|)$*. Thus, if block-GMRES is used to solve the correction equation, the inner iteration counts will not increase with the progress of the outer iteration.*

Proof The approximate solution to $A(\mathbb{P}^{(i)})^{-1}\tilde{Y} = BX^{(i)}$ in the k-th block-GMRES iteration starting with zero starting vector is

$$\tilde{Y}_k^{(i)} \in \operatorname{span}\{BX^{(i)}, A(\mathbb{P}^{(i)})^{-1}BX^{(i)}, ..., \big(A(\mathbb{P}^{(i)})^{-1}\big)^{k-1}BX^{(i)}\}. \tag{4.24}$$

It follows from (4.8) and $(\mathbb{P}^{(i)})^{-1}AX^{(i)} = X^{(i)}$ that

$$\begin{aligned} Y_1^{(i)} &= (\mathbb{P}^{(i)})^{-1}\tilde{Y}_1^{(i)} = (\mathbb{P}^{(i)})^{-1}BX^{(i)}F \\ &= (\mathbb{P}^{(i)})^{-1}\Big(AX^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)} - AV_2\big(S^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)} - M^{-1}S^{(i)}\big)\Big)F \\ &= X^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)}F + (\mathbb{P}^{(i)})^{-1}AV_2\big(M^{-1}S^{(i)} - S^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)}\big)F \\ &= X^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)}F + \Delta^{(i)}, \end{aligned} \tag{4.25}$$

where the jth column of $F \in \mathbb{C}^{p\times p}$ minimizes $\|BX^{(i)}e_j - A(\mathbb{P}^{(i)})^{-1}BX^{(i)}f\|$, i.e., the residual norm of the jth individual system (property of block-GMRES), and

$$\|\Delta^{(i)}\| \le \|(\mathbb{P}^{(i)})^{-1}AV_2\|\|F\|\|\bar{\mathcal{S}}_i\|\|S^{(i)}\|, \tag{4.26}$$

where the Sylvester operator $\bar{\mathcal{S}}_i : G \to \bar{\mathcal{S}}_i(G) = M^{-1}G - G(C^{(i)})^{-1}K^{-1}C^{(i)}$. Using the same derivation as in the proof of Proposition 4.2.2, we can show $\|\bar{\mathcal{S}}_i\| \to \|\bar{\mathcal{S}}\|$, where $\bar{\mathcal{S}} : G \to \bar{\mathcal{S}}(G) = M^{-1}G - GK^{-1}$. In addition, since $\mathcal{X}^{(i)} \to \mathcal{V}_1$ in (4.21), it follows that $\mathbb{P}^{(i)} \to \mathbb{P} \equiv P + (AV_1 - PV_1)V_1^*$. Thus $\|\Delta^{(i)}\| = O(\|S^{(i)}\|)$ is established.

We now investigate the residual norm $\|BX^{(i)} - AY_1^{(i)}\|$ of the linear system after Phase I of Algorithm 4.2. Recall the property of tuning that $\big(I - A(\mathbb{P}^{(i)})^{-1}\big)AX^{(i)} = 0$, and the property of block-GMRES that the approximate solution $\tilde{Y}_1^{(i)} \in \text{span}\{BX^{(i)}\}$. As block-GMRES minimizes the residual norm of each individual linear system of the block system, the j-th column of the block residual is

$$(4.27) \qquad \big\|\big(BX^{(i)} - A(\mathbb{P}^{(i)})^{-1}\tilde{Y}_1^{(i)}\big)e_j\big\| = \min_{f\in\mathbb{C}^p} \|BX^{(i)}e_j - A(\mathbb{P}^{(i)})^{-1}(BX^{(i)}f)\|$$
$$\leq \big\|\big(I - A(\mathbb{P}^{(i)})^{-1}\big)BX^{(i)}e_j\big\| \leq \big\|\big(I - A(\mathbb{P}^{(i)})^{-1}\big)BX^{(i)}\big\|$$
$$= \big\|\big(I - A(\mathbb{P}^{(i)})^{-1}\big)AV_2\big(S^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)} - M^{-1}S^{(i)}\big)\big\| \qquad \text{(see (4.8))}$$
$$\leq \big\|\big(I - A(\mathbb{P}^{(i)})^{-1}\big)AV_2\big\|\|\bar{\mathcal{S}}_i\|\|S^{(i)}\| = O(\|S^{(i)}\|),$$

from which follows

$$(4.28) \qquad \|BX^{(i)} - AY_1^{(i)}\| \leq \sum_{j=1}^{p} \left\|\big(BX^{(i)} - A(\mathbb{P}^{(i)})^{-1}\tilde{Y}_1^{(i)}\big)e_j\right\| = O(\|S^{(i)}\|).$$

Finally in Phase II of Algorithm 4.2, $Y_{k+1}^{(i)} = Y_1^{(i)} + dY_k^{(i)}$, where $dY_k^{(i)}$ is an approximate solution of the correction equation $A\,dY^{(i)} = BX^{(i)} - AY_1^{(i)}$. The stopping criterion requires that

$$(4.29) \qquad \frac{\|BX^{(i)} - AY_{k+1}^{(i)}\|}{\|BX^{(i)}\|} = \frac{\|BX^{(i)} - A(Y_1^{(i)} + dY_k^{(i)})\|}{\|BX^{(i)}\|}$$
$$= \frac{\|(BX^{(i)} - AY_1^{(i)}) - A\,dY_k^{(i)}\|}{\|BX^{(i)} - AY_1^{(i)}\|}\frac{\|BX^{(i)} - AY_1^{(i)}\|}{\|BX^{(i)}\|} \leq \delta \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)}).$$

Note that $\frac{\|(BX^{(i)}-AY_1^{(i)})-A\,dY_k^{(i)}\|}{\|BX^{(i)}-AY_1^{(i)}\|}$ is the relative residual norm of the correction equation for $dY_k^{(i)}$, and $\frac{\|BX^{(i)}-AY_1^{(i)}\|}{\|BX^{(i)}\|}$ is the relative residual norm of the original equation for $Y_1^{(i)}$. Therefore the prescribed stopping criterion of the inner iteration is satisfied if

$\frac{\|(BX^{(i)}-AY_1^{(i)})-A\,dY_k^{(i)}\|}{\|BX^{(i)}-AY_1^{(i)}\|}$ is bounded above by

$$\frac{\delta\|BX^{(i)}\|\sin\angle(A\mathcal{X}^{(i)},B\mathcal{X}^{(i)})}{\|BX^{(i)}-AY_1^{(i)}\|} \geq \frac{\delta\|BX^{(i)}\|\sigma^{(i)}\mathrm{sep}(T_{22},K^{(i)})}{\|\big(I-A(\mathbb{P}^{(i)})^{-1}\big)AV_2\|\|\bar{\mathcal{S}}_i\|\sqrt{1+\|Q\|^2}} \equiv \rho^{(i)}, \tag{4.30}$$

where we apply the lower bound on $\sin\angle(A\mathcal{X}^{(i)},B\mathcal{X}^{(i)})$ in (4.13) and the upper bound on $\|BX^{(i)}-AY_1^{(i)}\|$ in (4.28).

To study $\rho^{(i)}$, recall from the end of the proof of Proposition 4.2.2 that $\mathrm{sep}(T_{22},K^{(i)}) \to \mathrm{sep}(T_{22},K) > 0$, and $\sigma^{(i)} = \sigma_{min}(B)\|(X^{(i)*}A^*AX^{(i)})^{1/2}\| \to \sigma_{min}(B)\|(V_1^*A^*AV_1)^{1/2}\| > 0$. In addition, $\|\big(I-A(\mathbb{P}^{(i)})^{-1}\big)AV_2\| \to \|\big(I-A\mathbb{P}^{-1}\big)AV_2\|$ and $\|\bar{\mathcal{S}}_i\| \to \|\bar{\mathcal{S}}\|$. This means $\rho^{(i)}$, a lower bound on the relative tolerance for the correction equation, can be fixed independent of i. It then follows from Lemma 4.3.1 and our assumption concerning the decomposition (4.23) of $BX^{(i)}-AY_1^{(i)}$ that if block-GMRES is used to solve the correction equation, the inner iteration counts do not increase with the progress of the outer iteration. ∎

4.3.3 A general strategy for the phase I computation

It can be seen from Theorem 4.3.3 that the key to the success of the two-phase strategy is that in the first phase, an approximate solution $Y_1^{(i)}$ is obtained whose block residual norm $\|BX^{(i)}-AY_1^{(i)}\| = O(\|S^{(i)}\|)$. It is shown in Section 4.3.2 that such a $Y_1^{(i)}$ can be constructed inexpensively from a single step of block-GMRES with tuning applied to $AY^{(i)} = BX^{(i)}$. In fact, a valid $Y_1^{(i)}$ can also be constructed in other ways, in particular, by solving a set of least squares problems

$$\min_{f\in\mathbb{C}^p}\|BX^{(i)}e_j - AX^{(i)}f_j\| \qquad 1\leq j\leq p. \tag{4.31}$$

This is easily done using the QR factorization of $AX^{(i)}$. The solution f_j satisfies

$$\begin{aligned} \|BX^{(i)}e_j - AX^{(i)}f_j\| &= \min_{f\in\mathbb{C}^p} \|BX^{(i)}e_j - AX^{(i)}f\| \\ &\leq \|BX^{(i)}e_j - AX^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)}e_j\| \\ &= \left\|AV_2\left(S^{(i)}(C^{(i)})^{-1}K^{-1}C^{(i)} - M^{-1}S^{(i)}\right)e_j\right\| \qquad \text{(see (4.8))} \\ &\leq \|AV_2\|\|\bar{\mathcal{S}}_i\|\|S^{(i)}\| = O(\|S^{(i)}\|). \end{aligned} \tag{4.32}$$

Thus, with $Y_1^{(i)} = X^{(i)}[f_1, ..., f_p]$, it follows immediately that

$$\|BX^{(i)} - AY_1^{(i)}\| \leq \sum_{j=1}^{p} \|BX^{(i)}e_j - AX^{(i)}f_j\| \leq O(\|S^{(i)}\|), \tag{4.33}$$

so that the conclusion of Theorem 4.3.3 is also valid for this choice of $Y_1^{(i)}$.

This discussion reveals a connection between the two-phase strategy and the inverse correction method [67, 33] and the residual inverse power method [87], where the authors independently present essentially the same key idea for inexact inverse iteration. For example, [87] constructs $x^{(i+1)}$ by adding a small correction $z^{(i)}$ to $x^{(i)}$. Here, $z^{(i)}$ is the solution of $Az = \mu x^{(i)} - Ax^{(i)}$, where $\mu = x^{(i)*}Ax^{(i)}$ is the Rayleigh quotient, and $\mu x^{(i)} - Ax^{(i)}$ is the current eigenvalue residual vector that satisfies $\|\mu x^{(i)} - Ax^{(i)}\| = \min_{\alpha\in\mathbb{C}} \|\alpha x^{(i)} - Ax^{(i)}\|$. In Algorithm 4.2, we compute $Y_{k+1}^{(i)}$ by adding $dY_k^{(i)}$ to $Y_1^{(i)}$, where $dY_k^{(i)}$ is an approximate solution of $A\,dY^{(i)} = BX^{(i)} - AY_1^{(i)}$. Here $Y_1^{(i)}$ satisfies span$\{Y_1^{(i)}\} \approx$ span$\{X^{(i)}\}$ (see (4.25)), and $\|BX^{(i)} - AY_1^{(i)}\|$ is minimized by a single block-GMRES iteration. For both methods, the relative tolerance of the correction equation can be fixed independent of the outer iteration. The least squares formulation derived from (4.31) can be viewed as a generalization to subspace iteration of the residual inverse power method of [87].

Remark 3.4. In fact, all these approaches are also similar to what is done by the Jacobi-Davidson method. To be specific, the methods in [67, 33, 87] essentially compute a parameter β explicitly or implicitly such that $\|B(\beta x^{(i)}) - Ax^{(i)}\|$ is minimized or close to being minimized, then solve the correction equation $Az^{(i)} = B(\beta x^{(i)}) - Ax^{(i)}$ and get $x^{(i+1)}$ by normalizing $x^{(i)} + z^{(i)}$. The right hand side $B(\beta x^{(i)}) - Ax^{(i)}$ is identical or similar to that of the Jacobi-Davidson correction equation, i.e., the current eigenvalue residual vector. The difference is that the system solve required by the Jacobi-Davidson method forces the correction direction to be orthogonal to the current approximate eigenvector $x^{(i)}$. In addition, [26] shows that for inexact Rayleigh quotient iteration, solving the equation $(A - \sigma^{(i)}I)y^{(i)} = x^{(i)}$ ($\sigma^{(i)}$ is the Rayleigh quotient) with preconditioned full orthogonalization method (FOM) with *tuning* is equivalent to solving the simplified Jacobi-Davidson correction equation $(I - x^{(i)}x^{(i)*})(A - \sigma^{(i)}I)(I - x^{(i)}x^{(i)*})z^{(i)} = -(A - \sigma^{(i)})x^{(i)}$ with preconditioned FOM, as both approaches give the same inner iterate up to a constant.

4.4 Additional strategies to reduce inner iteration cost

In this section, we propose and study the use of deflation of converged Schur vectors, special starting vector for the correction equation, and iterative linear solvers with recycled subspaces to further reduce the cost of inner iteration.

4.4.1 Deflation of converged Schur vectors

With proper deflation of converged Schur vectors, we only need to apply matrix-vector products involving $\mathcal{A}$ to the unconverged Schur vectors. This reduces the inner iteration cost because the right hand side of the block linear system contains fewer columns. To successfully achieve this goal, two issues must be addressed: (1) how to simplify the procedure to detect converged Schur vectors and distinguish them from unconverged ones, and (2) how to apply tuning correctly to the block linear systems with reduced size, so that the relative tolerance of the correction equation can be fixed as in Theorem 4.3.3.

The first issue is handled by the Schur-Rayleigh-Ritz (SRR) procedure in Step

3 of Algorithm 4.1. The SRR step recombines and reorders the columns of $X^{(i+1)}$, so that its leading (leftmost) columns are approximate Schur vectors corresponding to the most dominant eigenvectors. Specifically, it forms the approximate Rayleigh quotient $\Theta^{(i)} = X^{(i)*}Y^{(i)} \approx X^{(i)*}\mathcal{A}X^{(i)}$, computes the Schur decomposition $\Theta^{(i)} = W^{(i)}T^{(i)}W^{(i)*}$ where the eigenvalues are arranged in *descending* order of magnitude in $T^{(i)}$, and orthogonalizes $Y^{(i)}W^{(i)}$ into $X^{(i+1)}$ (see Chapter 6 of [86]). As a result, the columns of $X^{(i+1)}$ will converge in order from left to right as the outer iteration proceeds. Then we only need to detect how many leading columns of $X^{(i+1)}$ have converged; the other columns are the unconverged Schur vectors.

To study the second issue, assume that $X^{(i)} = \big[X_a^{(i)}, X_b^{(i)}\big]$ where $X_a^{(i)}$ has converged. Then we deflate $X_a^{(i)}$ and solve the smaller block system $AY_b^{(i)} = BX_b^{(i)}$. When a single step of preconditioned block-GMRES with tuning is applied to this system (Phase I of Algorithm 4.2), it is important to *not* deflate $X_a^{(i)}$ in the tuned preconditioner (4.21). In particular, the effect of tuning (significant reduction of the linear residual norm in the first block-GMRES step) depends on the fact that $BX^{(i)}$ is an approximate invariant subspace of $A(\mathbb{P}^{(i)})^{-1}$. This nice property is valid only if we use the whole $X^{(i)}$ to define tuning.

To see this point, recall the partial Schur decomposition $B^{-1}AV_1 = V_1T_{11}$ in (4.1). We can further decompose this equality as

$$B^{-1}AV_1 = B^{-1}A\big[V_{1a},\ V_{1b}\big] = \big[V_{1a},\ V_{1b}\big]\begin{bmatrix} T_{11}^{\alpha} & T_{11}^{\beta} \\ 0 & T_{11}^{\gamma} \end{bmatrix} = V_1T_{11}. \tag{4.34}$$

It follows that $AV_{1b} = BV_{1a}T_{11}^{\beta} + BV_{1b}T_{11}^{\gamma}$, or equivalently

$$\left(-AV_{1a}(T_{11}^{\alpha})^{-1}T_{11}^{\beta} + AV_{1b}\right)(T_{11}^{\gamma})^{-1} = BV_{1b}. \tag{4.35}$$

In short, $\mathrm{span}\{BV_{1b}\} \subset \mathrm{span}\{AV_{1a}\} \cup \mathrm{span}\{AV_{1b}\}$, but $\mathrm{span}\{BV_{1b}\} \not\subseteq \mathrm{span}\{AV_{1b}\}$, because of the triangular structure of the partial Schur form in (4.34). If $X_a^{(i)} = V_{1a}$ is the set of converged dominant Schur vectors, and $X_b^{(i)} \approx V_{1b}$ is the set of unconverged Schur vectors, then these observations show that $B\mathcal{X}_b^{(i)}$ has considerable components in *both* $A\mathcal{X}_a^{(i)}$ and $A\mathcal{X}_b^{(i)}$. Therefore, when solving $AY_b^{(i)} = BX_b^{(i)}$, if we use *only* $X_b^{(i)}$ to define tuning in (4.21), so that $A\mathcal{X}_a^{(i)}$ is *not* an invariant subspace of $A(\mathbb{P}^{(i)})^{-1}$, then the right hand side $BX_b^{(i)}$ does not span an approximate invariant subspace of $A(\mathbb{P}^{(i)})^{-1}$. Thus the large one-step reduction of the linear residual norm (see (4.28)) will not occur, and many more block-GMRES iterations would be needed for the correction equation.

4.4.2 Special starting vector for the correction equation

The second additional means to reduce the inner iteration cost is to choose a good starting vector for the correction equation, so that the initial residual norm of the correction equation can be greatly reduced. We find that a good starting vector for the current equation can be constructed from a proper linear combination of the solutions of previously solved equations, because the right hand sides of several consecutive correction equations are close to being linearly dependent. Note that the feasibility of this construction of starting vector stems from the specific structure of the two-phase strategy: as tuning defined in (4.21) need not be applied in Phase II of Algorithm 4.2, the preconditioner does not depend on $X^{(i)}$, and thus we can work with preconditioned system matrices that are the same for the correction equation in all outer iterations.

To understand the effectiveness of this special starting vector, we need to see why the right hand sides of a few successive correction equations are close to being linearly dependent. Some insight can be obtained by analyzing the simple case with block size $p = 1$. To begin the analysis, consider using Algorithm 4.2 to solve $Ay^{(i)} = Bx^{(i)}$, where $x^{(i)} = v_1 c^{(i)} + V_2 S^{(i)}$ is the normalized current approximate eigenvector (see (4.5)). Here $c^{(i)}$ is a scalar and $S^{(i)} \in \mathbb{C}^{(n-1)\times 1}$ is a column vector.

In Phase I of Algorithm 4.2, we apply a single step of preconditioned GMRES to solve $A(\mathbb{P}^{(i)})^{-1}\tilde{y}^{(i)} = Bx^{(i)}$ and get $y_1^{(i)} = (\mathbb{P}^{(i)})^{-1}\tilde{y}_1^{(i)}$. Then $y_1^{(i)} = \alpha(\mathbb{P}^{(i)})^{-1}Bx^{(i)}$ where

$$\alpha = \text{argmin}_\alpha \|Bx^{(i)} - \alpha A(\mathbb{P}^{(i)})^{-1}Bx^{(i)}\| = \frac{(Bx^{(i)})^* A(\mathbb{P}^{(i)})^{-1}Bx^{(i)}}{\|A(\mathbb{P}^{(i)})^{-1}Bx^{(i)}\|^2} = \frac{\nu_\alpha}{\mu_\alpha}. \tag{4.36}$$

To evaluate α, noting that $K = \lambda_1$ in (4.2), we have from (4.8) that

$$Bx^{(i)} = \lambda_1^{-1}Ax^{(i)} - AV_2(\lambda_1^{-1}I - M^{-1})S^{(i)}. \tag{4.37}$$

From the tuning condition (4.21), $A(\mathbb{P}^{(i)})^{-1}(Ax^{(i)}) = Ax^{(i)}$, and therefore

$$A(\mathbb{P}^{(i)})^{-1}Bx^{(i)} = \lambda_1^{-1}Ax^{(i)} - A(\mathbb{P}^{(i)})^{-1}AV_2(\lambda_1^{-1}I - M^{-1})S^{(i)}. \tag{4.38}$$

To simplify the notation, let $J_1 = AV_2(\lambda_1^{-1}I - M^{-1})$ in (4.37) and $J_2 = A(\mathbb{P}^{(i)})^{-1}J_1$ in (4.38). Substituting (4.37) and (4.38) into (4.36), we get the numerator and denominator of α as follows:

$$\nu_\alpha = \lambda_1^{-2}\|Ax^{(i)}\|^2 - \lambda_1^{-1}(Ax^{(i)})^*(J_1 + J_2)S^{(i)} + S^{(i)*}J_1^*J_2S^{(i)}, \tag{4.39}$$

and

$$\mu_\alpha = \lambda_1^{-2}\|Ax^{(i)}\|^2 - 2\lambda_1^{-1}(Ax^{(i)})^*J_2S^{(i)} + S^{(i)*}J_2^*J_2S^{(i)}. \tag{4.40}$$

Both ν_α and μ_α are quadratic functions of $S^{(i)}$. As $\|S^{(i)}\| \to 0$, we use the first order approximation of $\alpha = \nu_\alpha/\mu_\alpha$ (neglecting higher order terms in $S^{(i)}$) to get

$$\alpha \approx 1 + \frac{\lambda_1}{\|Ax^{(i)}\|^2}(Ax^{(i)})^*(J_2 - J_1)S^{(i)}. \tag{4.41}$$

Assume that $M = U_M\Lambda_M U_M^{-1}$ where $\Lambda_M = \text{diag}(\lambda_2, \lambda_3, ..., \lambda_n)$ and M has normalized columns. It follows that $V_2U_M = \begin{bmatrix} v_2, v_3, ..., v_n \end{bmatrix}$ are the normalized eigenvectors of (A, B) and $AV_2U_M = BV_2U_M\Lambda_M$. Suppose $x^{(0)} = c_1v_1 + \sum_{k=2}^n c_kv_k$. If the system $Ay^{(i)} = Bx^{(i)}$ is solved exactly for all i, then $x^{(i)} = c_1v_1 + \sum_{k=2}^n c_k(\lambda_1/\lambda_k)^iv_k$ up to some constant scaling factor. On the other hand, we know from (4.5) that $x^{(i)} = v_1c^{(i)} + V_2S^{(i)} = v_1c^{(i)} + V_2U_M(U_M^{-1}S^{(i)})$. Therefore,

$$U_M^{-1}S^{(i)} = \Big(c_2\Big(\frac{\lambda_1}{\lambda_2}\Big)^i, c_3\Big(\frac{\lambda_1}{\lambda_3}\Big)^i, ..., c_n\Big(\frac{\lambda_1}{\lambda_n}\Big)^i\Big)^* \tag{4.42}$$

up to some constant. If $\{|\lambda_2|, |\lambda_3|, ..., |\lambda_l|\}$ are tightly clustered and well-separated from $\{|\lambda_{l+1}|, ..., |\lambda_n|\}$, the magnitudes of the first $l-1$ entries of $U_M^{-1}S^{(i)}$ are significantly bigger than those of the other entries.

With (4.37), (4.38), (4.41) and (4.42), using the first order approximation again, we can write the right hand side of the correction equation as

$$\begin{aligned} Bx^{(i)} - Ay_1^{(i)} &= Bx^{(i)} - \alpha A(\mathbb{P}^{(i)})^{-1}Bx^{(i)} \\ \approx &\Big(I - \frac{Ax^{(i)}}{\|Ax^{(i)}\|}\Big(\frac{Ax^{(i)}}{\|Ax^{(i)}\|}\Big)^*\Big)\big(A(\mathbb{P}^{(i)})^{-1} - I\big)A(V_2U_M)(\lambda_1^{-1}I - \Lambda_M^{-1})(U_M^{-1}S^{(i)}), \end{aligned} \tag{4.43}$$

where the vector $(V_2U_M)(\lambda_1^{-1}I - \Lambda_M^{-1})(U_M^{-1}S^{(i)})$ is dominated by a linear combination of $\{v_2, ..., v_l\}$. In addition, as $x^{(i)} \rightarrow v_1$, the first matrix factor in the second line of (4.43) filters out the component of Av_1. As a result, for big enough i, the right hand side of the current correction equation is roughly a linear combination of those of $l-1$ previous consecutive equations.

Using the above observation, a starting vector $dY_0^{(i)}$ for the correction equation can be constructed as follows:

$$dY_0^{(i)} = \mathrm{dY}_{l-1}y, \quad \text{where } \mathrm{dY}_{l-1} = \big[dY^{(i-l+1)}, dY^{(i-l+2)}, ..., dY^{(i-1)}\big], \text{ and} \tag{4.44}$$

$$\begin{aligned} &y = \mathrm{argmin}_{y\in\mathbb{C}^{l-1}}\big\|\mathrm{RHS}_{l-1}y - \big(BX^{(i)} - AY_1^{(i)}\big)\big\|, \\ &\text{where} \quad \mathrm{RHS}_{l-1} = \big[BX^{(i-l+1)} - AY_1^{(i-l+1)}, ..., BX^{(i-1)} - AY_1^{(i-1)}\big]. \end{aligned} \tag{4.45}$$

In practice, we find that $l = 3$ or 4 is enough to generate a good starting vector. The cost of solving this small least squares problem (4.45) is negligible.

4.4.3 Linear solvers with recycled subspaces

In Phase II of Algorithm 4.2, we need to solve the correction equation $A\,dY^{(i)} = BX^{(i)} - AY_1^{(i)}$. The third strategy to speed up the inner iteration is to use linear solvers with subspace recycling to solve the sequence of correction equations. This methodology is specifically designed to efficiently solve a long sequence of *slowly-changing* linear systems. After the iterative solution of one linear system, a small set of vectors from the current subspace for the candidate solutions is carefully selected and the space spanned by these vectors is "recycled", i.e., used for the iterative solution of the next linear system. The cost of solving subsequent linear systems can usually be reduced by subspace recycling, because the iterative solver does not have to build the subspace for the candidate solution from scratch. A typical solver of this type is the Generalized Conjugate Residual with implicit inner Orthogonalization and Deflated Restarting (GCRO-DR) in [61], which was developed using ideas for the solvers with special truncation [14] and restarting [55] for a single linear system.

In [61], the preconditioned system matrix changes from one linear system to the next, and the recycled subspace taken from the previous system must be transformed by matrix-vector products involving the current system matrix to fit into the solution of the current system (see the Appendix of [61]). In the setting of solving the sequence of correction equations, fortunately, this transformation cost can be avoided with Algorithm 4.2, because the preconditioned system matrix is the same for the correction equation in all outer iterations.

We implement a block version of GCRO-DR to solve the correction equation. The block generalization is very similar to the extension of GMRES to block-GMRES. The residual norm of the block linear system is minimized in each block iteration over all candidate solutions in the union of the recycled subspace and a block Krylov subspace (see [61] for details). The dimension of the recycled subspace can be chosen independent of the block size. The authors of [61] suggest choosing the harmonic Ritz vectors corresponding to smallest harmonic Ritz values for the recycled subspaces. The harmonic Ritz vectors are approximate "smallest" eigenvectors of the preconditioned system matrix. If they do approximate these eigenvectors reasonably well, this choice tends to reduce the duration of the initial latency of GMRES convergence, which is typically observed when the system matrix has a few eigenvalues of very small magnitude; see [20]. We also include dominant Ritz vectors in the recycled subspace, as suggested in [61]. As our numerical experiments show (see Section 4.5), when the use of harmonic Ritz vectors fails to reduce the inner iteration cost, the set of dominant Ritz vectors is still a reasonable choice for subspace recycling.

4.5 Numerical experiments

In this section, we test the effectiveness of the strategies described in Sections 4.3 and 4.4 for solving the block linear systems arising in inexact subspace iteration. We show that the two-phase strategy (Algorithm 4.2) achieves performance similar to that achieved when tuning is used at *every* block-GMRES step (the approach given in [61]): both methods keep the inner iteration cost from increasing, though the required tolerance for the solve decreases progressively. The numerical experiments also corroborate the analysis that a single block-GMRES iteration with tuning reduces the linear residual norm to a small quantity proportional to $\|S^{(i)}\|$, so that the relative tolerance of the correction equation remains a moderately small constant independent of $\|S^{(i)}\|$. We have also seen experimentally that the least squares strategy of Section 4.3.3 achieves the same effect. The Phase I step is somewhat more expensive using tuned preconditioned GMRES than the least squares approach, but for the problems we studied, the former approach required slightly fewer iterations in Phase II, and the total of inner iterations is about the same for the two methods. For the sake of brevity, we only present the results obtained by the two-phase strategy where tuning is applied in Phase I.

We also show that deflation gradually decreases the inner iteration cost as more converged Schur vectors are deflated. In addition, the use of subspace recycling and special starting vector lead to further reduction of inner iteration counts.

We first briefly explain the criterion to detect the convergence of Schur vectors in Step 3 of Algorithm 4.1. Let $I_{p,j} = (I_j \ 0)^T \in \mathbb{R}^{p\times j}$ so that $X^{(i)} I_{p,j}$ contains the first j columns of $X^{(i)}$. Right after the SRR step, we find the largest integer j for which the following criterion is satisfied:

$$\|BX^{(i)}I_{p,j} - AX^{(i)}I_{p,j}T_j^{(i)}\| \leq \|BX^{(i)}I_{p,j}\|\epsilon, \tag{4.46}$$

where $T_j^{(i)}$ is the $j \times j$ leading block of $T^{(i)}$ coming from the SRR step (see Section 4.1 for details). If (4.46) holds for j but not for $j+1$, we conclude that exactly j Schur vectors have converged and should be deflated. This stopping criterion is analogous to that of the EB12 function (subspace iteration) of HSL (formerly the Harwell Subroutine Library) [41, 48].

We use four test problems. The first one is MHD4800A/B from Matrix Market [51], a real matrix pencil of order 4800 which describes the Alfvén spectra in magnetohydrodynamics (MHD). We use the shift-invert operator $\mathcal{A} = (A - \sigma B)^{-1}B$ with σ close to the left end of the spectrum. Since it is very hard to find a preconditioner for A, we use the ILU preconditioner for $A - \sigma B$ with drop tolerance 1.5×10^{-7} given by MATLAB's `ilu`. Using MATLAB's `nnz` to count the number of nonzero entries, we have $\texttt{nnz}(A - \sigma B) = 120195$, and $\texttt{nnz}(L) + \texttt{nnz}(U) = 224084$. In fact, a slightly bigger tolerance, say 1.75×10^{-7}, leads to failure of `ilu` due to a zero pivot.

The second problem is UTM1700A/B from Matrix Market, a real matrix pencil of size 1700 arising from a tokamak model in plasma physics. We use Cayley transformation to compute the leftmost eigenvalues $\lambda_{1,2} = -0.032735 \pm 0.3347i$ and

$\lambda_3 = 0.032428$. Note that $\Im(\lambda_{1,2})$ is 10 times bigger than λ_3, and there are some real eigenvalues to the right of λ_3 with magnitude smaller than $\Im(\lambda_{1,2})$. We choose the ILU preconditioner with drop tolerance 0.001 for $A - \sigma_1 B$.

Problems 3 and 4 come from the linear stability analysis of a model of two-dimensional incompressible fluid flow over a backward facing step, constructed using the IFISS software package [17, 18]. The domain is $[-1, L] \times [-1, 1]$, where $L = 15$ in Problem 3 and $L = 22$ in Problem 4; the Reynolds numbers are 600 and 1200 respectively. Let u and v be the horizontal and vertical components of the velocity, p be the pressure, and ν the viscosity. The boundary conditions are as follows:

$$\begin{aligned}
&u = 4y(1-y),\ v = 0 \text{ (parabolic inflow)} && \text{on } x = -1, y \in [0,1];\\
&\nu\frac{\partial u}{\partial x} - p = 0,\ \frac{\partial v}{\partial y} = 0 \text{ (natural outflow)} && \text{on } x = L, y \in [-1,1];\\
&u = v = 0 \text{ (no-slip)} && \text{on all other boundaries.}
\end{aligned} \tag{4.47}$$

We use a biquadratic/bilinear (Q_2-Q_1) finite element discretization with element width $\frac{1}{16}$ (grid parameter 6 in the IFISS code). The sizes of the two problems are 72867 and 105683 respectively. Block linear solves are done using the least squares commutator preconditioner [19]. For Problems 3 and 4, we try both shift-invert (subproblem (a)) and Cayley transformation (subproblem (b)) to detect a small number of critical eigenvalues.

	p	k	σ (σ_1)	σ_2	δ	ϵ	l_1	l_2
Prob 1	9	7	-370	–	2×10^{-5}	5×10^{-11}	5	10
Prob 2	3	3	-0.0325	0.125	1×10^{-5}	5×10^{-11}	5	10
Prob 3(a)	7	7	0	–	1×10^{-3}	5×10^{-10}	0	20
Prob 3(b)	5	3	0	-0.46	1×10^{-4}	5×10^{-10}	0	20
Prob 4(a)	5	4	0	–	1×10^{-3}	5×10^{-10}	0	30
Prob 4(b)	4	4	0	-0.24	5×10^{-4}	5×10^{-10}	0	30

Table 4.1: Parameters used to solve the test problems

For completeness, we summarize all parameters used in the solution of each test problem in Table 4.1. These parameters are chosen to deliver approximate eigenpairs of adequate accuracies, show representative behavior of each solution strategy, and keep the total computational cost moderate.

1. p, k – we use $X^{(i)}$ with p columns to compute k eigenpairs of (A, B)
2. $\sigma, \sigma_1, \sigma_2$ – the shifts of $\mathcal{A} = (A - \sigma B)^{-1}B$ and $\mathcal{A} = (A - \sigma_1 B)^{-1}(A - \sigma_2 B)$
3. δ – the relative tolerance for solving $AY^{(i)} = BX^{(i)}$ is $\delta \sin\angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})$
4. ϵ – the error in the convergence test (4.46)
5. l_1, l_2 – we use l_1 harmonic Ritz vectors corresponding to harmonic Ritz values of smallest magnitude and l_2 dominant Ritz vectors for subspace recycling

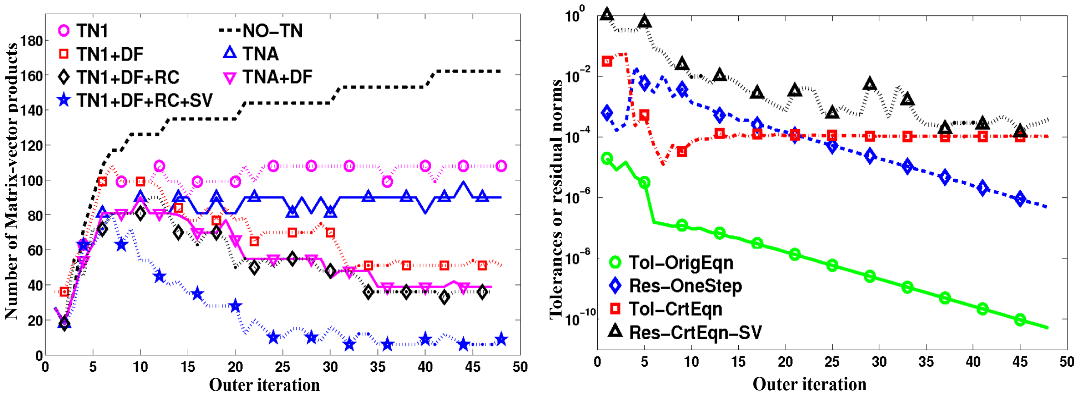

Figure 4.1: Performance of different solution strategies for Problem 1 (a): preconditioned matrix-vector product counts of the inner iteration against the outer iteration (b): behavior of the two-phase strategy and starting vector.

The performance of different strategies to solve $AY^{(i)} = BX^{(i)}$ for each problem is shown in Figures 4.1–4.4. We use Problem 1 as an example to explain the results. In Figure 4.1(a), the preconditioned matrix-vector product counts of the inner iteration are plotted against the progress of the outer iteration. The curves with different markers correspond to solution strategies as follows:

1. "NO-TN" (no marker with dotted line) – Solve $AY^{(i)} = BX^{(i)}$ by preconditioned block-GMRES *without* tuning.

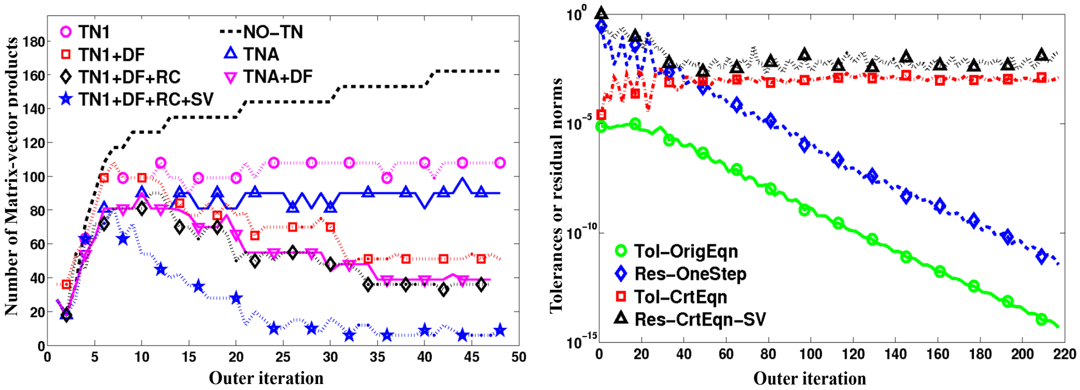

Figure 4.2: Performance of different solution strategies for Problem 2 (a): preconditioned matrix-vector product counts of the inner iteration against the outer iteration (b): behavior of the two-phase strategy and starting vector.

2. "TNA" (△ marker with solid line) – Solve $AY^{(i)} = BX^{(i)}$ by preconditioned block-GMRES *with* tuning.

3. "TNA+DF" (▽ marker with solid line) – Apply "TNA" and deflation of converged Schur vectors.

4. "TN1" (○ marker with dashed line) – Solve $AY^{(i)} = BX^{(i)}$ by Algorithm 4.2, without any additional enhancements.

5. "TN1+DF" (□ marker with dashed line) – Apply "TN1" and deflation of converged Schur vectors.

6. "TN1+DF+RC" (◇ marker with dashed line) – Apply "TN1+DF" and use GCRO-DR to solve the correction equation in Phase II of Algorithm 4.2.

7. "TN1+DF+RC+SV" (★ marker wish dashed line) – Apply "TN1+DF+RC" and use the special starting vector for the correction equation.

From Figure 4.1(a), we see that if no tuning is used, the matrix-vector product counts in each outer iteration increase gradually to over 160, whereas they are fixed at 110 if the two-phase strategy (without any additional enhancements) is applied (◯ TN1). If converged Schur vectors are deflated, the matrix-vector product counts decrease gradually from 100 to 80, 70, and finally to about 50 (□ TN1+DF). The use of recycled subspace of dimension 15 further reduces the counts by approximately 15 (♢ TN1+DF+RC). The special starting vector makes an additional significant improvement: the counts are reduced to less than 20 after the 23rd outer iteration, and even to less than 10 after the 35th outer iteration (★ TN1+DF+RC+SV).

Figure 4.1(b) plots four quantities against the outer iteration as follows:

1. "Tol-OrigEqn" (○ marker) – The relative tolerance $\delta \sin \angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})$ for the original linear system $AY^{(i)} = BX^{(i)}$.
2. "Res-OneStep" (♢ marker) – The relative residual norm $\frac{\|BX^{(i)} - AY_1^{(i)}\|}{\|BX^{(i)}\|}$ after one step block-GMRES iteration with tuning (Phase I of Algorithm 4.2).
3. "Tol-CrtEqn" (□ marker) – The relative tolerance $\frac{\delta \|BX^{(i)}\| \sin \angle(A\mathcal{X}^{(i)}, B\mathcal{X}^{(i)})}{\|BX^{(i)} - AY_1^{(i)}\|}$ for the correction equation $A\,dY^{(i)} = BX^{(i)} - AY_1^{(i)}$.
4. "Res-CrtEqn-SV" (△ marker) – Given the starting vector $dY_0^{(i)}$, the initial relative residual norm $\frac{\|(BX^{(i)} - AY_1^{(i)}) - A\,dY_0^{(i)}\|}{\|BX^{(i)} - AY_1^{(i)}\|}$ of the correction equation.

It is clear from Figure 4.1(b) that one step of block-GMRES with tuning reduces the residual norm of $AY^{(i)} = BX^{(i)}$ to a small quantity proportional to $\|S^{(i)}\|$ ("Res-OneStep"), so that the relative tolerance of the correction equation ("Tol-CrtEqn") is approximately a constant. In addition, the special starting vector $dY_0^{(i)}$ considerably reduces the initial residual norm of the correction equation. For example, in the 45th outer iteration, the relative tolerance for $AY^{(45)} = BX^{(45)}$ is $\delta \sin \angle(A\mathcal{X}^{(45)}, B\mathcal{X}^{(45)}) \approx 10^{-10}$; a single block-GMRES iteration with tuning decreases the relative linear residual norm to $\frac{\|BX^{(45)} - AY_1^{(45)}\|}{\|BX^{(45)}\|} \approx 10^{-6}$, so that $dY_k^{(45)}$ for the correction equation only needs to satisfy $\frac{\|(BX^{(45)} - AY_1^{(45)}) - A\,dY_k^{(45)}\|}{\|BX^{(45)} - AY_1^{(45)}\|} \le 10^{-4}$; see (4.29) for details. Moreover, the starting vector $dY_0^{(45)}$ makes $\frac{\|(BX^{(45)} - AY_1^{(45)}) - A\,dY_0^{(45)}\|}{\|BX^{(45)} - AY_1^{(45)}\|}$ almost as small as 10^{-4}, so that little additional effort is needed to solve the correction equation.

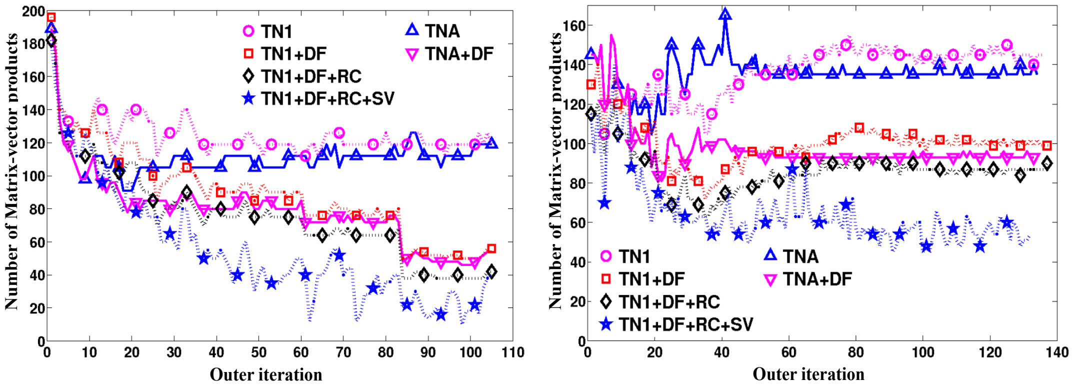

Figure 4.3: Performance of different solution strategies for Problem 3(a) and 3(b): preconditioned matrix-vector product counts of the inner iteration against the outer iteration.

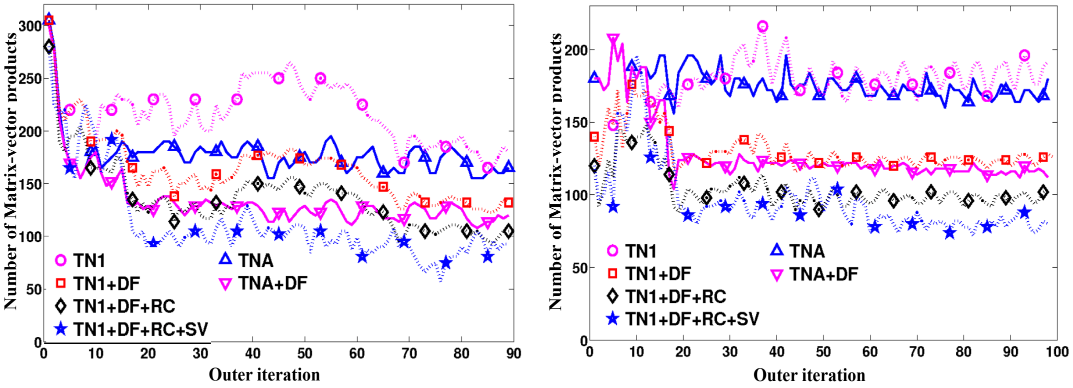

Figure 4.4: Performance of different solution strategies for Problems 4(a) and 4(b): preconditioned matrix-vector product counts of the inner iteration against the outer iteration.

Table 4.2 shows the number of preconditioned matrix-vector products when different strategies are used to solve the block linear systems $AY^{(i)} = BX^{(i)}$ for each problem. For Problems 1 and 2, we achieve a speed up ratio of 3.4 and 4.4 respectively by the combined use of all strategies, compared to the original use of tuning ("TNA") proposed in [65]; for Problems 3(a) and 3(b), we reduce the inner iteration cost by over 50%; for Problem 4(a) and 4(b), the savings are 36% and 45% respectively. Recall from (4.22) that tuning requires an application of the Sherman-Morrison-Woodbury formula in each *inner* iteration. The two-phase strategy uses tuning only in one block-GMRES iteration and hence avoids the overhead of tuning. The additional strategies of Section 4.4 only entail computation of the recycled subspaces and the

	NO-TN	TNA	TN1	TNA +DF	TN1 +DF	TN1+DF +RC	**TN1+DF +RC+IG**
Prob 1	6435	3942	4761	2606	3254	2498	1175
Prob 2	19110	12183	15357	10854	13886	7220	2744
Prob 3(a)	–	11704	13097	8270	9370	7863	5785
Prob 3(b)	–	17475	18600	12613	13792	11806	8521
Prob 4(a)	–	15785	19350	11978	14578	12183	10100
Prob 4(b)	–	17238	17468	12624	12892	10197	9428

Table 4.2: Number of preconditioned matrix-vector products for different solution strategy for each problem

starting vector (both costs are small) for the block system in each *outer* iteration.

One can see from Figures 4.1–4.4 that the two-phase strategy without subspace recycling and special starting vector generally requires slightly more inner iterations than the original tuned version of the solves (compare "TN1" with "TNA" and "TN1+DF" with "TNA+DF"). The reason is that the tuned version of a preconditioner $\mathbb{P}^{(i)}$ has two possible advantages over its untuned version P:

1. With a tuned preconditioner, the right hand side of $A(\mathbb{P}^{(i)})^{-1}\tilde{Y}^{(i)} = BX^{(i)}$ is an approximate invariant subspace of the preconditioned operator $A(\mathbb{P}^{(i)})^{-1}$.
2. In addition, $A(\mathbb{P}^{(i)})^{-1}$ typically has more favorable properties, such as better eigenvalue clustering, for Krylov subspace methods than AP^{-1}.

The first advantage is the original motivation for the use of tuning, as studied in [25, 26, 65] and this chapter. The second one is studied in [28] for solving linear systems that arise when inexact Arnoldi method is applied to compute a few smallest eigenvalues of a matrix from Matrix Market. We attribute the slight increase in inner iteration counts associated with Algorithm 4.2 to its use of untuned preconditioners in the second phase. However, with Algorithm 4.2, the overhead of tuning is avoided, and further reduction of inner iteration counts can be achieved by using subspace recycling (no transformation of subspaces needed) and special starting vectors.

Moreover, our experience suggests that the second advantage of tuning tends to be less of a factor if the untuned preconditioner P is very strong (most eigenvalues of AP^{-1} are clustered around 1). For instance, for Problem 1, compared to the strategy “TNA” where tuning is used in every inner iteration, the two-phase strategy “TN1” requires about 18 more preconditioned matrix-vector products (or a 20% relative increase) for each block linear system after the 20th outer iteration; see Figure 4.1(a). Similarly for Problem 2, “TN1” needs about 15 more matrix-vector multiplications (or a 25% relative increase) than “TNA” for each system after the 75th outer iteration. However, for Problems 3(a), 3(b) and 4(b), the relative increase is only about 10% in the last tens of outer iterations; for Problem 4(a), though “TNA” obviously outperforms “TN1” in the first 67 outer iterations, the relative difference between the two approaches still falls far below 20% in the last 22 outer iterations. The reason is that the “clustering” effect of tuning is more pronounced when the relatively weak ILU preconditioners are used in Problems 1 and 2, and is less influential for Problems 3 and 4 where the strong least square commutator preconditioner [20] is used.

In all numerical experiments, deflation of converged Schur vectors always reduces the *preconditioned matrix-vector product* counts, but the inner *iteration* counts tends to increase slightly. This agrees with our experience with the behavior of block linear solvers. For instance, if it takes 10 block iterations to solve a block system with 8 right hand sides to some tolerance, then it usually takes more than 10 but less than 20 block iterations to solve the system with block size 4 to the some tolerance.

We successfully reduce some inner iteration cost by using block GCRO-DR (subspace recycling). However, a conclusive evaluation of the effectiveness of this approach is beyond the scope of this chapter. To the best of our knowledge, block GCRO-DR has not been mentioned in the literature. The dimensions of the recycled subspaces we use in block GCRO-DR are commensurate with those used in single-vector GCRO-DR [61]. Since block GCRO-DR generally needs much bigger subspaces to extract candidate solutions than its single-vector counterpart, it might be beneficial to use recycled subspaces of bigger dimensions. In addition, the harmonic Ritz vectors corresponding to smallest harmonic Ritz values are not necessarily a good choice for recycling if, for example, the smallest eigenvalues of the preconditioned system matrix are not well-separated from other eigenvalues [13]. We speculate this is the case in Problems 3 and 4, where there are several very small eigenvalues and some small ones when the least squares commutator preconditioner is used (see [20]). In this case, it is the dominant Ritz vectors that are useful.

4.6 Concluding remarks

We have studied inexact subspace iteration for solving generalized non-Hermitian eigenvalue problems with shift-invert and Cayley transformations. We provide new perspectives on tuning and discuss the connection of the two-phase strategy to the inverse correction method, the residual inverse power method and the Jacobi-Davidson method. The two-phase strategy applies tuning only in the first block-GMRES iteration and solves the correction equation with a fixed relative tolerance. It prevents the inner iteration counts from increasing as the outer iteration proceeds, as the original approach in [65] does. Three additional strategies are studied to further reduce the inner iteration cost, including deflation, subspace recycling and special initial guess. Numerical experiments show clearly that the combined use of all these techniques leads to significant reduction of inner iteration counts.

5 Inexact implicitly restarted Arnoldi method

In this chapter, we study an inexact implicitly restarted Arnoldi (IRA) method for computing a few eigenpairs of generalized non-Hermitian eigenvalue problems with spectral transformation, where in each Arnoldi step (outer iteration) the matrix-vector product involving the transformed operator is performed by iterative solution (inner iteration) of the corresponding linear system of equations. We provide new perspectives and analysis of two major strategies that help reduce the inner iteration cost: the preconditioner with tuning and a sequence of gradually relaxed tolerances for the solution of the linear systems. We study a new tuning strategy constructed from vectors in both previous and the current IRA cycles, and show how the tuning is used in a new two-phase algorithm to greatly reduce inner iteration counts. We give an upper bound of the allowable tolerances of the linear systems and propose an alternative estimate of the tolerances. In addition, the inner iteration cost can be further reduced through the use of subspace recycling with iterative linear solvers.

5.1 Introduction

In Chapter 4, we studied an inexact subspace iteration for computing several non-dominant eigenpairs of generalized non-Hermitian eigenvalue problems $Av = \lambda Bv$. This algorithm works with subspaces of fixed dimensions larger than one, and converges linearly if the inner solve is carried out with reasonable accuracy. In many cases, however, the linear convergence may not be satisfactory as it entails a large number of outer iterations. In addition, the inner solve performed by block-GMRES or block GCRO-DR may require a prohibitive amount of storage.

In this chapter, we investigate an inexact implicitly restarted Arnoldi method (IRA) with spectral transformation to find a few non-extremal eigenpairs of $Av =$

λBv. Both the convergence rate and storage requirement can be improved by the use of inexact IRA. Specifically, IRA works with subspaces of increasing dimensions and usually converges superlinearly in practice. Furthermore, this method may have less demand of storage because the inner solve is performed for a linear system with a single right-hand side. The primary concern of this chapter, similar to that of Chapters 3 and 4, is to study some critical techniques for efficient iterative solution of the linear systems that arise when the inexact IRA is used to solve eigenvalue problems.

In the past decade, considerable developments have been made in understanding inexact projection-based eigenvalue algorithms, such as the Lanczos and the Arnoldi methods. It was found in [35] and [8] that the matrix-vector product must be computed accurately in the initial Lanczos or Arnoldi steps, but the accuracy can be relaxed as the algorithm proceeds without obviously affecting the convergence of approximate eigenpairs. An analysis of this phenomenon is given in [74] for the Arnoldi method, using perturbation theory of invariant subspaces. It is shown there that the allowable errors of matrix-vector products in Arnoldi steps should be inversely proportional to the eigenvalue residual norm of the desired eigenpair. Therefore, as the Arnoldi method proceeds and converges to the eigenpair of interest, the accuracy of matrix-vector products can be relaxed. The use of inexact matrix-vector products has also been studied in the setting of Krylov subspace linear solvers; see [9, 10],[77, 78, 79] and [82, 92].

Further study of inexact Arnoldi methods is given in [28], where the tuning strategy and the relaxed accuracy of matrix-vector products are extended to inexact IRA with shift-invert transformation for standard eigenvalue problems. For the linear systems arising in Arnoldi steps (outer iterations) in a given IRA cycle, tuning is developed using all available Arnoldi vectors in that cycle. Numerical experiments show that for a test problem from Matrix Market [51], an ILU preconditioner with this tuning considerably reduces the inner iteration counts. It is observed there and confirmed in this chapter that this improvement is mainly due to the fact that tuning helps cluster the eigenvalues of the preconditioned system matrix of the linear system in each Arnoldi step. In addition, [28] proposes a practical estimate of the allowable relaxed tolerances for the solution of the linear systems, using the distance between the spectra of two matrices containing the wanted and unwanted Ritz values to replace the separation [86] between the two. Numerical experiments show that the total inner iteration counts of inexact IRA can be substantially reduced by the combined use of tuning and relaxed tolerances.

In this chapter, we refine the tuning strategy and further study the allowable tolerances for inner solves of the inexact IRA method for generalized non-Hermitian eigenvalue problems. We first study a new tuning strategy constructed for a given Arnoldi step using the solutions of linear systems in previous Arnoldi steps. In addition, we propose a two-phase strategy to solve the linear system in the current Arnoldi step. Specifically, we apply only one step of preconditioned GMRES with tuning to the current linear system to get an approximate solution, then solve the correction equation with any appropriate preconditioned linear solver; in particular, tuning is not needed for the correction equation. We show that the approximate solution obtained in the first phase can be a very good one if enough solution vectors from previous Arnoldi steps are used. With this special approximate solution, the correction equation can be solved with a relative tolerance much larger than that of the original linear system, and inner iteration counts can hence be reduced considerably. In addition, we use a special type of iterative linear solver with subspace recycling to solve the sequence of correction equations as the IRA method proceeds. We show that subspace recycling is cheap to use in this setting and can further reduce the inner iteration counts substantially.

A second goal of this chapter is to present a refined analysis of the allowable tolerance for the linear systems in the inexact IRA method. We first give an upper bound of the allowable tolerance, showing that violation of this bound necessarily leads to contamination of the desired approximate invariant subspace by excessive errors of inner solves. We then give a theoretically more accurate estimate of the allowable tolerance, which is between the upper bound and a conservative lower bound from [28]. As this estimate contains information not available until the end of the current IRA cycle, we use a computable substitute obtained at the end of the previous IRA cycle. We then compare this heuristic estimate with that from [28] and discuss the impact of the accuracy of the estimate on the inner solves.

This chapter is organized as follows. In Section 2, we briefly review spectral transformations, the IRA method and some properties of the algorithm when exact shifts (unwanted Ritz values) are used in filter polynomials. We discuss a few strategies for the inner solves in Section 3, studying the properties of the new tuning strategy and the new two-phase strategy for solving the linear system in each Arnoldi step. We also explain the effectiveness of the linear solver with subspace recycling applied to solve the correction equations. In Section 4, we study the allowable tolerances of the linear systems and give an necessary upper bound for the tolerance. A new heuristic estimate of the allowable tolerance is proposed and used in numerical experiments to corroborate the accuracy of the estimate from [28]. Numerical experiments in Section 5 show that the combined use of the new tuning, subspace recycling and relaxed tolerances greatly reduces the total inner iteration counts. In Section 6 we make some concluding remarks.

5.2 Review: the implicitly restarted Arnoldi method

To make the exposition smooth, we briefly review the implicitly restarted Arnoldi (IRA) method. IRA was developed by Sorensen [85] in 1992 and is a most well-known breakthrough in the area of eigenvalue computation. This robust algorithm has been implemented in ARPACK [49], a mathematical software package of high quality which has become the standard solver for large non-Hermitian eigenvalue problems.

The key technique of the IRA method is the implicit application of a filter polynomial to a given Arnoldi decomposition to produce the effect of several steps of a restarted Arnoldi computation without any matrix-vector multiplications. Specifically, at the end of the ith IRA cycle we have an m-step Arnoldi decomposition

$$\mathcal{A}U_m^{(i)} = U_m^{(i)} H_m^{(i)} + h_{m+1,m}^{(i)} u_{m+1}^{(i)} e_m^T. \tag{5.1}$$

Suppose $\kappa_1, \kappa_2, ..., \kappa_{m-k} \in \mathbb{C}$ are estimates of $m-k$ eigenvalues of $\mathcal{A}$ obtained from this process corresponding to a part of the spectrum we are not interested in. We then use these numbers as shifts to apply $m-k$ shifted QR steps to $H_m^{(i)}$ and get a Krylov decomposition

$$\mathcal{A}\tilde{U}_m^{(i)} = \tilde{U}_m^{(i)} \tilde{H}_m^{(i)} + h_{m+1,m}^{(i)} u_{m+1}^{(i)} (e_m^T Q^{(i)}), \tag{5.2}$$

where $Q^{(i)} = Q_1 Q_2 ... Q_{m-k}$ is the product of $m-k$ upper Hessenberg unitary matrices, $\tilde{U}_m^{(i)} = U_m^{(i)} Q^{(i)}$, $\tilde{H}_m^{(i)} = Q^{(i)*} H_m^{(i)} Q^{(i)}$ is upper Hessenberg, and $e_m^T Q^{(i)}$ is the last row of $Q^{(i)}$ with $k-1$ zero leading entries. For details, see [85], or [32, 86].

The restarted Arnoldi decomposition is then obtained from the first k columns of the above Krylov decomposition as follows

$$\begin{aligned} \mathcal{A}\tilde{U}_k^{(i)} &= \tilde{U}_k^{(i)} \tilde{H}_k^{(i)} + \tilde{h}_{k+1,k}^{(i)} \tilde{u}_{k+1}^{(i)} e_k^T + (h_{m+1,m}^{(i)} q_{m,k}^{(i)}) u_{m+1}^{(i)} e_k^T, \qquad \text{or,} \\ \mathcal{A}U_k^{(i+1)} &= U_k^{(i+1)} H_k^{(i+1)} + h_{k+1,k}^{(i+1)} u_k^{(i+1)} e_k^T. \end{aligned} \tag{5.3}$$

Here $q_{m,k}^{(i)}$ is the (m,k) entry of $Q^{(i)}$, $U_k^{(i+1)} = \tilde{U}_k^{(i)}$, and $H_k^{(i+1)} = \tilde{H}_k^{(i)}$. Note that both $\tilde{u}_{k+1}^{(i)}$ and $u_{m+1}^{(i)}$ are orthogonal to $U_k^{(i+1)}$. Let $\hat{u}_k^{(i+1)} = \tilde{h}_{k+1,k}^{(i)} \tilde{u}_{k+1}^{(i)} + (h_{m+1,m}^{(i)} q_{m,k}^{(i)}) u_{m+1}^{(i)}$, then $h_{k+1,k}^{(i+1)} = \|\hat{u}_k^{(i+1)}\|$, and $u_k^{(i+1)} = (h_{k+1,k}^{(i+1)})^{-1} \hat{u}_k^{(i+1)}$. Clearly, no additional matrix-vector product involving $\mathcal{A}$ is used for the restart. For the restarted Arnoldi decomposition, it can be shown that $u_1^{(i+1)} = (\mathcal{A} - \kappa_1 I)(\mathcal{A} - \kappa_2 I)...(\mathcal{A} - \kappa_{m-k} I)u_1^{(i)}$ up to a constant scaling factor. In other words, the eigenvector component corresponding to the unwanted spectrum in $u_1^{(i)}$ is filtered out by the filter polynomial.

An inexact implicitly restarted Arnoldi method is given in Algorithm 5.1.

Algorithm 5.1 inexact implicitly restarted Arnoldi (IRA) method

```
Given a normalized $u_1^{(0)} \in \mathbb{C}^n$, $j = 1$
for $i = 0, 1, 2, ...$ until convergence do
    1. Compute $\mathcal{A}u_j^{(i)}$ by solving $Ay = Bu_j^{(i)}$, so that the approximate solution
    $y_{k+1}$ satisfies $\frac{\|Bu_j^{(i)} - Ay_{k+1}\|}{\|Bu_j^{(i)}\|} \leq \delta^{(i,j)}$.
    Here $\delta^{(i,j)}$ is a relative tolerance specified or computed by some means.
    2. Expand the Arnoldi decomposition by orthogonalizing $y$ against $u_1^{(i)}, ..., u_j^{(i)}$
    and normalizing; the new Arnoldi decomposition is $\mathcal{A}U_j^{(i)} = U_j^{(i)} H_j^{(i)} + h_{j+1,j}^{(i)} u_{j+1}^{(i)} e_j^T$.
    $j \leftarrow j + 1$.
    3. If $j = m$, test for convergence.
    If not converged, invoke the implicit restart procedure to get
    $\mathcal{A}U_k^{(i+1)} = U_k^{(i+1)} H_k^{(i+1)} + h_{k+1,k}^{(i+1)} u_{k+1}^{(i+1)} e_k^T$, and $j \leftarrow k + 1$.
end for
```

In this study, we choose the "exact shifts" strategy for the IRA method, which uses the unwanted eigenvalues of $H_m^{(i)}$ (Ritz values) as shifts for the implicit restart. This is the default choice in ARPACK and has proved successful in many applications. Some properties of the IRA method with the exact shifts strategy are given as follows.

Proposition 5.2.1 (Corollary 2.3, Chapter 5 of [86]) *Suppose $\mu_1, ..., \mu_m$ are eigenvalues of $H_m^{(i)}$. If the implicit QR steps are performed with shifts $\{\mu_{k+1}, \mu_{k+2}, ..., \mu_m\}$, then*

$$\tilde{H}_m^{(i)} = Q^{(i)*} H_m^{(i)} Q^{(i)} = \begin{bmatrix} \tilde{H}_k^{(i)} & \tilde{H}_m^{12(i)} \\ 0 & \tilde{H}_m^{22(i)} \end{bmatrix}, \tag{5.4}$$

where $\tilde{H}_m^{22(i)}$ is an upper triangular matrix with $\mu_{k+1}, \mu_{k+2}, ..., \mu_m$ on its diagonal.

The proposition shows that $\tilde{h}_{k+1,1}^{(i)} = 0$ if exact shifts are used. This observation immediately leads to the following result.

Proposition 5.2.2 *Let the Schur decomposition of* $H_m^{(i)}$ *be* $H_m^{(i)} = W_m^{(i)} T_m^{(i)} W_m^{(i)*}$, *where* $W_m^{(i)} = \left[W_m^{1(i)},\ W_m^{2(i)} \right]$ *is unitary, and*

$$T_m^{(i)} = \begin{bmatrix} T_m^{11(i)} & T_m^{12(i)} \\ 0 & T_m^{22(i)} \end{bmatrix} \tag{5.5}$$

with $\lambda(T_m^{11(i)}) = \{\mu_1, \mu_2, ..., \mu_k\}$, $\lambda(T_m^{22(i)}) = \{\mu_{k+1}, \mu_{k+2}, ..., \mu_m\}$, *and* $\lambda(T_m^{11(i)}) \cap \lambda(T_m^{22(i)}) = \emptyset$. *Then*

$$\begin{aligned} (5.6) \qquad \|\mathcal{A}U_m^{(i)} W_m^{1(i)} - U_m^{(i)} W_m^{1(i)} T_m^{11(i)}\| &= \|\mathcal{A}U_k^{(i+1)} - U_k^{(i+1)} H_k^{(i+1)}\|, \text{ and} \\ (5.7) \qquad \|h_{m+1,m}^{(i)} u_{m+1}^{(i)} e_m^T W_m^{1(i)}\| &= \|h_{k+1,k}^{(i+1)} u_{k+1}^{(i+1)} e_k^T\|. \end{aligned}$$

Proof Let $Q^{(i)} = \left[Q^{1(i)},\ Q^{2(i)} \right]$. From (5.4) and (5.5) we have $Q^{1(i)*} H_m^{(i)} Q^{1(i)} = \tilde{H}_k^{(i)}$ and $W_m^{1(i)*} H_m^{(i)} W^{1(i)} = T_m^{11(i)}$. Since $\lambda(\tilde{H}_k^{(i)}) = \lambda(T_m^{11(i)}) = \{\mu_1, \mu_2, ..., \mu_k\}$, there exists a $k \times k$ unitary matrix $V^{(i)}$ such that $V^{(i)*} \tilde{H}_k^{(i)} V^{(i)} = T_m^{11(i)}$ and $W_m^{1(i)} = Q^{1(i)} V^{(i)}$. Note from (5.2) and (5.3) that $U_m^{(i)} Q^{1(i)} = U_k^{(i+1)}$. Therefore

$$\begin{aligned} (5.8) \qquad & \|\mathcal{A}U_m^{(i)} W_m^{1(i)} - U_m^{(i)} W_m^{1(i)} T_m^{11(i)}\| \\ = \ & \|\mathcal{A}U_m^{(i)} Q^{1(i)} V^{(i)} - U_m^{(i)} Q^{1(i)} V^{(i)} T_m^{11(i)} V^{(i)*} V^{(i)}\| \\ = \ & \|(\mathcal{A}U_k^{(i+1)} - U_k^{(i+1)} \tilde{H}_k^{(i)}) V^{(i)}\| = \|\mathcal{A}U_k^{(i+1)} - U_k^{(i+1)} \tilde{H}_k^{(i)}\|. \end{aligned}$$

Since $\tilde{h}_{k+1,k}^{(i)} = 0$, we have $h_{k+1,k}^{(i+1)} u_{k+1}^{(i+1)} e_k^T = (h_{m+1,m}^{(i)} q_{m,k}^{(i)}) u_{m+1}^{(i)} e_k^T$ from (5.3), and therefore

$$\begin{aligned} (5.9) \qquad & \|h_{m+1,m}^{(i+1)} u_{m+1}^{(i)} e_m^T W_m^{1(i)}\| = \|h_{m+1,m}^{(i+1)} u_{m+1}^{(i)} e_m^T Q^{1(i)} V^{(i)}\| \\ = \ & \|(h_{m+1,m}^{(i+1)} q_{m,k}^{(i)}) u_{m+1}^{(i)} e_k^T\| = \|h_{k+1,k}^{(i+1)} u_{k+1}^{(i+1)} e_k^T\|. \end{aligned}$$

■

For the exact IRA method (where the matrix-vector products involving $\mathcal{A}$ are computed exactly), (5.7) can be derived from (5.6). For inexact IRA, however, the "true eigenvalue residuals" in (5.6) and the "estimated eigenvalue residuals" in (5.7) are different. Proposition 5.2.2 shows that the two types of eigenvalue residual norms are "restart-invariant" if exact shifts are used: both quantities at the end of the ith IRA cycle are the same as those at the beginning of the $(i+1)$-th IRA cycle.

We have so far reviewed the derivation and some properties of the ordinary IRA method. For generalized eigenvalue problems with singular B, an alternative B-orthogonal IRA has been studied and recommended in the literature. For generalized Hermitian problems with positive (semi)definite B, the original motivation of the B-orthogonal Lanczos method is to preserve symmetry of the shift-invert operator $\mathcal{A} = (A - \sigma B)^{-1}B$ with respect to the B-inner product (see Section 4.2, Chapter 6 of [86]). However if B is only semidefinite, the B-orthogonal Lanczos method may suffer from severe growth of null-space errors as the Lanczos method proceeds. Reference [59] presents a strategy to purge Ritz vectors of null-space errors. For generalized non-Hermitian problems with semidefinite B, [52] observes and analyzes similar growth of null-space errors in the B-orthogonal Arnoldi method and proposes an implicit restart with a zero-shift to purge the null-space errors. A recent study in [66] considers a type of generalized non-Hermitian problems with saddle point structure and shows that the ordinary IRA applied to an equivalent problem of reduced size is much less vulnerable to null-space errors. This observation is corroborated by [90], where the author suggests using the ordinary IRA to solve non-Hermitian problems with singular B. In this chapter, we use the ordinary IRA for eigenvalue problems arising in linear stability analysis (where B is symmetric positive semidefinite) and successfully obtain approximate eigenpairs of accuracies close to machine precision.

5.3 New strategies for solving linear systems in inexact IRA

To improve the efficiency for solving the linear systems arising in inexact eigenvalue algorithms, a special type of preconditioner with "tuning" is studied in [25, 26, 65, 28]. Given an existing preconditioner P, tuning constructs a special low-rank update of P such that the tuned preconditioner $\mathbb{P}$ behaves like the system matrix A on a certain set of vectors X. It is shown in these papers that the inner iteration counts needed to solve the linear system preconditioned by $\mathbb{P}$ are substantially smaller than those required to solve the system preconditioned by P.

For example, consider inexact subspace iteration with $\mathcal{A} = A^{-1}$ used to detect a few smallest eigenvalues of A. In each outer iteration, we approximately solve the block linear system $AY^{(i)} = X^{(i)}$, where $X^{(i)}$ contains the current approximate Schur vectors (therefore $X^{(i)*}X^{(i)} = I$). It is shown in [65] that a decreasing sequence of tolerances for the block systems is necessary to guarantee the linear convergence of $X^{(i)}$ to the desired invariant subspace. As a result, the block-GMRES iteration counts required to solve $AP^{-1}\tilde{Y}^{(i)} = X^{(i)}$ (with $Y^{(i)} = P^{-1}\tilde{Y}^{(i)}$) increases gradually as the outer iteration progresses. To resolve this difficulty, P is replaced by the tuned preconditioner defined as

$$\mathbb{P}^{(i)} = P + (A - P)X^{(i)}X^{(i)*}, \tag{5.10}$$

for which $\mathbb{P}^{(i)}X^{(i)} = AX^{(i)}$, or equivalently, $A(\mathbb{P}^{(i)})^{-1}(AX^{(i)}) = AX^{(i)}$. That is, $AX^{(i)}$ spans an invariant subspace of the preconditioned system matrix with tuning corresponding to eigenvalue 1. The authors show that for $A(\mathbb{P}^{(i)})^{-1}\tilde{Y}^{(i)} = X^{(i)}$, the right hand side $X^{(i)}$ spans an approximate invariant subspace of $A(\mathbb{P}^{(i)})^{-1}$, and the block-GMRES iteration counts needed for solving this preconditioned system do *not* increase with the outer iteration progress.

The tuning strategy is extended in [28] to an inexact IRA method for standard eigenvalue problems. Let m and k be the order of the Arnoldi decomposition, i.e., the number of columns in the Hessenberg matrix right before and after the implicit restart. Assume after the jth ($0 \leq j \leq m-k-1$) Arnoldi step in the ith IRA cycle, an Arnoldi decomposition $\mathcal{A}U_{k+j}^{(i)} = U_{k+j}^{(i)}H_{k+j}^{(i)} + h_{k+j+1,k+j}^{(i)}u_{k+j+1}^{(i)}e_{k+j}^T$ is already computed, and $Ay = u_{k+j+1}^{(i)}$ needs to be solved in the (j+1)-th Arnoldi step. The authors define the tuned preconditioning matrix as $\mathbb{P}_{k+j+1}^{(i)} = P + (A - P)XX^*$, where $X = U_{k+j+1}^{(i)}$ contains the Arnoldi vectors in the ith IRA cycle. They find that inner iteration counts required to solve $A(\mathbb{P}_{k+j+1}^{(i)})^{-1}\tilde{y} = u_{k+j+1}^{(i)}$ are smaller than those needed to solve $AP^{-1}\tilde{y} = u_{k+j+1}^{(i)}$, because $A(\mathbb{P}_{k+j+1}^{(i)})^{-1}$ has better eigenvalue clustering than that of AP^{-1}. This "clustering" effect of tuning is quite different from the original motivation of this strategy studied in [25, 26, 65]. In particular, $u_{k+j+1}^{(i)}$ is generally not a very good approximate eigenvector of $A(\mathbb{P}_{k+j+1}^{(i)})^{-1}$.

In this section, we propose and study a new tuning strategy for solving the linear systems of equations that arise in inexact IRA for generalized non-Hermitian eigenvalue problems. To study the new tuning strategy under ideal conditions, we assume in this section that the linear system in each Arnoldi step is solved accurately (to machine precision). We will see from numerical experiments in Section 5 that the property of tuning under ideal conditions still holds approximately when the inner solves are performed inexactly, except for the last few IRA cycles when the allowable tolerances are relaxed significantly. We show how this tuning can be used in a new two-phase algorithm to solve the linear systems in each Arnoldi step. In addition, we discuss the use of subspace recycling with iterative solvers in the second phase of the two-phase algorithm.

5.3.1 The new tuning strategy

The motivation for the new tuning strategy is similar to that discussed in [25, 26, 65]. The aim to make the right hand side of the linear system in the current Arnoldi step an approximate eigenvector of the preconditioned system matrix, so that the inner iteration counts can be greatly reduced. Suppose we are in the ith IRA cycle and already have $\mathcal{A}U_{k+j}^{(i)} = U_{k+j}^{(i)}H_{k+j}^{(i)} + h_{k+j+1,k+j}^{(i)}u_{k+j+1}^{(i)}e_{k+j}^T$. We then compute $\mathcal{A}u_{k+j+1}^{(i)}$ by solving $Ay = Bu_{k+j+1}^{(i)}$ during the current step. Recall that for a given X with orthonormal columns, the tuned preconditioner $\mathbb{P} = P + (A-P)XX^*$ satisfies $\mathbb{P}X = AX$, i.e., $A\mathbb{P}^{-1}(AX) = AX$. The motivation behind tuning requires that X be chosen so that the right-hand side $Bu_{k+j+1}^{(i)}$ of the current linear system approximately lies in the subspace spanned by AX, an invariant subspace of $A\mathbb{P}^{-1}$.

Consider the following choice of X:

$$X_p^{(i,l)} = \left[\mathcal{A}U_m^{(i-l)}, \mathcal{A}U_{k+1:m}^{(i-l+1)}, ..., \mathcal{A}U_{k+1:m}^{(i-1)}, \mathcal{A}U_{k+1:k+j}^{(i)}\right], \tag{5.11}$$

where $U_{k+1:m}^{(r)}$ stands for the $(k+1)$-th through the mth columns of $U_m^{(r)}$, and $p = (m-k)(l-1) + m + j$ is the number of the vectors in $X_p^{(i,l)}$. We refer to $X_p^{(i,l)}$ as the set of "solution vectors", because its columns are solutions of the linear systems in previous Arnoldi steps. For example, the first vector in $X_p^{(i,l)}$ is $\mathcal{A}u_1^{(i-l)}$, the solution of $Ay = Bu_1^{(i-l)}$ in the first step of the $(i-l)$-th IRA cycle. Note that this system may not literally be solved in practice due to the implicit restart.

Let $U_p^{(i,l)} = \left[U_m^{(i-l)}, U_{k+1:m}^{(i-l+1)}, ..., U_{k+1:k+j}^{(i)}\right]$ and $\mathcal{U}_p^{(i,l)} = \text{span}\{U_p^{(i,l)}\}$, $\mathcal{X}_p^{(i,l)} = \text{span}\{X_p^{(i,l)}\}$, $A\mathcal{X}_p^{(i,l)} = B\mathcal{U}_p^{(i,l)} = \text{span}\{BU_m^{(i-l)}, BU_{k+1:m}^{(i-l+1)}, ..., BU_{k+1:k+j}^{(i)}\}$. In the following derivation, we use the calligraphic letter to stand for the subspaces spanned by some set of *column vectors* denoted by the same letter in Roman fonts. For instance, $\mathcal{U}_{k+j}^{(i)} = \text{span}\{U_{k+j}^{(i)}\}$. To study the relation between $Bu_{k+j+1}^{(i)}$ and $A\mathcal{X}_p^{(i,l)}$, we begin with the following lemma, which shows that the range of $U_p^{(i,l)}$ is a Krylov subspace.

Lemma 5.3.1 *Suppose IRA does not break down. Then* $\mathcal{U}_p^{(i,l)} = \mathcal{K}_p(\mathcal{A}, u_1^{(i-l)})$.

Proof First, $\mathcal{U}_{m+1}^{(i-l)} = \mathcal{K}_{m+1}(\mathcal{A}, u_1^{(i-l)})$. Since $u_{k+1}^{(i-l+1)}$ is a linear combination of $u_{m+1}^{(i-l)}$ and $\tilde{u}_{k+1}^{(i-l)} \in \mathcal{U}_m^{(i-l)}$ (see (5.3)), $\text{span}\{U_m^{(i-l)}, u_{k+1}^{(i-l+1)}\} = \mathcal{K}_{m+1}(\mathcal{A}, u_1^{(i-l)})$ holds.

As we have orthogonalized $\mathcal{A}u_{k+1}^{(i-l+1)}$ against $\mathcal{U}_{k+1}^{(i-l+1)} \subset \operatorname{span}\{U_m^{(i-l)}, u_{k+1}^{(i-l+1)}\}$ (note that $\mathcal{U}_k^{(i-l+1)} \subset \mathcal{U}_m^{(i-l)}$; see (5.3)) to get $u_{k+2}^{(i-l+1)}$, $\operatorname{span}\{U_m^{(i-l)}, u_{k+1}^{(i-l+1)}, u_{k+2}^{(i-l+1)}\} = \mathcal{K}_{m+2}(\mathcal{A}, u_1^{(i-l)})$ follows. Similar reasoning holds for all following Arnoldi vectors if IRA does not break down, and the theorem is established. ∎

The angle between a vector v and a subspace $\mathcal{U}$ (denoted as $\angle(v,\mathcal{U})$) is defined as the angle between v and the orthogonal projection of v onto $\mathcal{U}$. Obviously, $v \in \mathcal{U}$ if and only if $\angle(v,\mathcal{U}) = 0$. Therefore, $Bu_{k+j+1}^{(i)}$ approximately lies in $A\mathcal{X}_p^{(i,l)} = B\mathcal{U}_p^{(i,l)}$ if and only if $\angle(Bu_{k+j+1}^{(i)}, B\mathcal{U}_p^{(i,l)})$ is small, and this small angle condition holds if $\varphi_p^{(i)} = \angle(u_{k+j+1}^{(i)}, \mathcal{U}_p^{(i,l)})$ is small. The following theorem suggests that, given the starting vector $\mathcal{A}u_1^{(i-l)}$ of $X_p^{(i,l)}$, $\varphi_p^{(i)}$ can be small enough for large p, because it decreases linearly with p when $p > m$.

Theorem 5.3.2 *Let $\varphi_{p-1}^{(i)} = \angle(u_{k+j}^{(i)}, \mathcal{U}_{p-1}^{(i,l)})$ $(1 \le j \le m-k)$, so that $u_{k+j}^{(i)} = u_{p-1}\cos\varphi_{p-1}^{(i)} + u_{p-1}^{\perp}\sin\varphi_{p-1}^{(i)}$, where $u_{p-1} \in \mathcal{U}_{p-1}^{(i,l)}$ and $u_{p-1}^{\perp} \perp \mathcal{U}_{p-1}^{(i,l)}$ are unit vectors. Let the orthogonal projection of $\|\mathcal{A}u_{p-1}^{\perp}\|^{-1}\mathcal{A}u_{k+j}^{(i)}$ onto $\mathcal{U}_p^{(i,l)}$ be $w_p\eta_p$, where $w_p \in \mathcal{U}_p^{(i,l)}$ is a unit vector, and let $\alpha_p = \angle(\mathcal{A}u_{p-1}^{\perp}, \mathcal{U}_p^{(i,l)})$, and $\beta_p = \angle(w_p, \mathcal{U}_{k+j}^{(i)})$. Then*

$$(5.12) \qquad \tan\varphi_p^{(i)} = \nu_p \sin\varphi_{p-1}^{(i)}, \quad \text{where } \nu_p = \frac{\sin\alpha_p}{\eta_p \sin\beta_p}. \qquad (1 \le j \le m-k)$$

(Note that α_p, β_p, η_p and ν_p all depend on the IRA cycle number i. To simplify the notation, we omit the superscripts for these scalars.)

Proof Let $\rho = \frac{\|\mathcal{A}u_{p-1}\|}{\|\mathcal{A}u_{p-1}^{\perp}\|}$. Since $\mathcal{U}_p^{(i,l)} = \mathcal{K}_p(\mathcal{A}, u_1^{(i-l)})$, we have

$$(5.13) \qquad \begin{aligned} \mathcal{A}u_{k+j}^{(i)} &= \mathcal{A}u_{p-1}\cos\varphi_{p-1}^{(i)} + \mathcal{A}u_{p-1}^{\perp}\sin\varphi_{p-1}^{(i)} \\ &= \rho\|\mathcal{A}u_{p-1}^{\perp}\| w_{p1}\cos\varphi_{p-1}^{(i)} + \|\mathcal{A}u_{p-1}^{\perp}\|(w_{p2}\cos\alpha_p + w_p^{\perp}\sin\alpha_p)\sin\varphi_{p-1}^{(i)} \\ &= \|\mathcal{A}u_{p-1}^{\perp}\|(w_{p1}\rho\cos\varphi_{p-1}^{(i)} + w_{p2}\cos\alpha_p\sin\varphi_{p-1}^{(i)} + w_p^{\perp}\sin\alpha_p\sin\varphi_{p-1}^{(i)}) \\ &= \|\mathcal{A}u_{p-1}^{\perp}\|(w_p\eta_p + w_p^{\perp}\sin\alpha_p\sin\varphi_{p-1}^{(i)}), \end{aligned}$$

where w_{p1},w_{p2},$w_p \in \mathcal{U}_p^{(i,l)}$ and $w_p^{\perp} \perp \mathcal{U}_p^{(i,l)}$ are unit vectors, and $w_p \eta_p = w_{p1} \rho \cos \varphi_{p-1}^{(i)} + w_{p2} \cos \alpha_p \sin \varphi_{p-1}^{(i)}$ is the orthogonal projection of $\frac{\mathcal{A}u_{k+j}^{(i+l)}}{\|\mathcal{A}u_{p-1}^{\perp}\|}$ onto $\mathcal{U}_p^{(i,l)}$. It follows immediately that $\tan \angle(\mathcal{A}u_{k+j}^{(i)}, \mathcal{U}_p^{(i,l)}) = \frac{\sin \alpha_p \sin \varphi_{p-1}^{(i)}}{\eta_p}$.

We then orthogonalize $\mathcal{A}u_{k+j}^{(i)}$ against $\mathcal{U}_{k+j}^{(i)} \subset \mathcal{U}_p^{(i,l)}$ to get $u_{k+j+1}^{(i)}$. Let $\mathcal{U}_{k+j}^{(i)\perp}$ be the orthogonal complement of $\mathcal{U}_{k+j}^{(i)}$ in $\mathcal{U}_p^{(i,l)}$. Then $w_p = w_{p3} \cos \beta_p + w_{p4} \sin \beta_p$, where $w_{p3} \in \mathcal{U}_{k+j}^{(i)}$ and $w_{p4} \in \mathcal{U}_{k+j}^{(i)\perp}$ are unit vectors, and $\beta_p = \angle(w_p, \mathcal{U}_{k+j}^{(i)})$. Orthogonalizing $\mathcal{A}u_{k+j}^{(i)}$ against $\mathcal{U}_{k+j}^{(i)}$ removes the w_{p3} component from w_p, so that $u_{k+j+1}^{(i)}$ equals $w = w_{p4} \eta_p \sin \beta_p + w_p^{\perp} \sin \alpha_p \sin \varphi_{p-1}^{(i)}$ up to a constant scaling factor. It follows that $\tan \angle(u_{k+j+1}^{(i)}, \mathcal{U}_p^{(i,l)}) = \frac{\sin \alpha_p \sin \varphi_{p-1}^{(i)}}{\eta_p \sin \beta_p}$, and (5.12) is established. ▮

Remark 3.1. In Theorem 5.3.2 we are interested in the non-trivial case where $l > 0$. If $l = 0$, then $p = k + j$, and $\mathcal{U}_p^{(i,0)} = \mathcal{U}_{k+j}^{(i)}$. Therefore $\beta_p = \angle(w_p, \mathcal{U}_{k+j}^{(i)}) = 0$ (because $w_p \in \mathcal{U}_p^{(i,0)}$ by definition), ν_p is infinity, and $\varphi_p^{(i)} = \pi/2$. This is consistent with the fact that Arnoldi vectors in the same IRA cycle are orthogonal.

Remark 3.2. We have assumed that exact shifts are used for the implicit restart. In this case $u_{k+1}^{(i)} = u_{m+1}^{(i-1)}$; see (5.3) and Proposition 5.2.1. Therefore Theorem 5.3.2 also holds for $j = 0$, with $u_{k+j}^{(i)}$ replaced by $u_m^{(i-1)}$.

It is obvious from (5.12) that $\varphi_p^{(i)}$ decreases linearly with p if ν_p remains a constant smaller than 1. In practice ν_p is not a constant, but we have solid empirical evidences that $\varphi_p^{(i)}$ does decrease with p linearly.

Theorem 5.3.2 shows that with the choice of $X_p^{(i,l)}$ in (5.11), $\angle(u_{k+j+1}^{(i)}, \mathcal{U}_p^{(i,l)})$ is small for large enough p. As a result, $\angle(Bu_{k+j+1}^{(i)}, B\mathcal{U}_p^{(i,l)}) = \angle(Bu_{k+j+1}^{(i)}, A\mathcal{X}_p^{(i,l)})$ is also small. In other words, $Bu_{k+j+1}^{(i)}$ is an approximate eigenvector of $A\mathbb{P}^{-1}$ (where the tuned preconditioner $\mathbb{P}$ is constructed using $X_p^{(i,l)}$), because it approximately lies in $A\mathcal{X}_p^{(i,l)}$, an invariant subspace of $A\mathbb{P}^{-1}$. In the following subsection, we show how this observation can be used in a new two-phase algorithm for solving $Ay = Bu_{k+j+1}^{(i)}$.

5.3.2 A two-phase strategy to solve the linear systems in Arnoldi steps

With the new tuning discussed in subsection 5.3.1, we now propose a new two-phase algorithm for efficiently solving $Ay = Bu^{(i)}_{k+j+1}$ in Step 1 of Algorithm 5.1. This algorithm provides new insights into the use of tuning and its properties.

Algorithm 5.2 Two-phase strategy for solving $Ay = Bu^{(i)}_{k+j+1}$

1. Construct tuning using (5.11) and apply a *single* step of preconditioned GMRES with tuning to get an approximate solution y_1.
2. For a given ϵ, choose either a fixed tolerance $\delta = \delta_f(\epsilon)$, or a relaxed tolerance $\delta = \delta_r(\epsilon)$ by some means. Solve the *correction equation* $Az = Bu^{(i)}_{k+j+1} - Ay_1$ with *any* appropriate preconditioned iterative solver to get an approximate correction z_q, such that the corrected iterate $y_{q+1} = y_1 + z_q$ satisfies $\frac{\|Bu^{(i)}_{k+j+1} - Ay_{q+1}\|}{\|Bu^{(i)}_{k+j+1}\|} \leq \delta$, or equivalently, the correction z_q satisfies $\frac{\|(Bu^{(i)}_{k+j+1} - Ay_1) - Az_q\|}{\|Bu^{(i)}_{k+j+1} - Ay_1\|} \leq \frac{\delta \|Bu^{(i)}_{k+j+1}\|}{\|Bu^{(i)}_{k+j+1} - Ay_1\|}$.

In particular, tuning need *not* be used to solve the correction equation. Thus we can work with a fixed preconditioned system matrix for the correction equation in all Arnoldi steps.

Let $u^{(i)}_{k+j+1} = u_p c^{(i,l)}_p + u^{\perp}_p s^{(i,l)}_p$, where $u_p \in \mathcal{U}^{(i,l)}_p$ and $u^{\perp}_p \perp \mathcal{U}^{(i,l)}_p$ are unit vectors, $c^{(i,l)}_p$ and $s^{(i,l)}_p$ are the cosine and sine of $\angle(u^{(i)}_{k+j+1}, \mathcal{U}^{(i,l)}_p)$. We have shown by Theorem 5.3.2 that $s^{(i,l)}_p$ can be small enough for large p. The analysis of Algorithm 5.2 is given in the following major theorem.

Theorem 5.3.3 *Suppose Algorithm 5.2 is used to solve $Ay = Bu^{(i)}_{k+j+1}$. Then Phase I of Algorithm 5.2 gives $y_1 = \mathcal{A} u_p c^{(i,l)}_p + O(s^{(i,l)}_p)$ (up to a constant scaling factor) and the corresponding relative residual norm $\frac{\|Bu^{(i)}_{k+j+1} - Ay_1\|}{\|Bu^{(i)}_{k+j+1}\|} = O(s^{(i,l)}_p)$. Consequently, the stopping criterion of Algorithm 5.2 is satisfied if and only if the relative residual of the correction equation $\frac{\|(Bu^{(i)}_{k+j+1} - Ay_1) - Az_q\|}{\|Bu^{(i)}_{k+j+1} - Ay_1\|} \leq \frac{\delta \|Bu^{(i)}_{k+j+1}\|}{\|Bu^{(i)}_{k+j+1} - Ay_1\|} = \frac{\delta}{O(s^{(i,l)}_p)}$.*

Proof It is shown in Section 5.3.1 that if the preconditioning matrix $\mathbb{P}^{(i,l)}_p$ is constructed using $X^{(i,l)}_p$, then $A(\mathbb{P}^{(i,l)}_p)^{-1}(AX^{(i,l)}_p) = AX^{(i,l)}_p$. That is, $A\mathcal{X}^{(i,l)}_p = B\mathcal{U}^{(i,l)}_p$ is an invariant subspace of dimension p of $A(\mathbb{P}^{(i,l)}_p)^{-1}$ corresponding to eigenvalue 1. It follows that for $u_p \in \mathcal{U}_p^{(i,l)}$, $(\mathbb{P}_p^{(i,l)})^{-1}(Bu_p) = A^{-1}(Bu_p)$.

The approximate solution to $A(\mathbb{P}_p^{(i,l)})^{-1}\tilde{y} = Bu_{k+j+1}^{(i)}$ after one step of GMRES iteration is $\tilde{y}_1 \in \operatorname{span}\{Bu_{k+j+1}^{(i)}\}$, i.e., $\tilde{y}_1 = \gamma Bu_{k+j+1}^{(i)}$ with some scalar γ. Therefore

$$\begin{aligned} y_1 &= (\mathbb{P}_p^{(i,l)})^{-1}\tilde{y}_1 = (\mathbb{P}_p^{(i,l)})^{-1}(\gamma Bu_{k+j+1}^{(i)}) = \gamma(\mathbb{P}_p^{(i,l)})^{-1}(Bu_p c_p^{(i,l)} + Bu_p^{\perp} s_p^{(i,l)}) \\ &= \gamma\left(A^{-1}Bu_p c_p^{(i,l)} + (\mathbb{P}_p^{(i,l)})^{-1}Bu_p^{\perp} s_p^{(i,l)}\right) = \gamma(\mathcal{A}u_p c_p^{(i,l)} + \delta_p^{(i,l)}), \end{aligned} \tag{5.14}$$

where $\|\delta_p^{(i,l)}\| = s_p^{(i,l)}\|(\mathbb{P}_p^{(i,l)})^{-1}Bu_p^{\perp}\| = O(s_p^{(i,l)})$.

Now consider the residual norm after one step of GMRES with tuning:

$$\begin{aligned} &\|Bu_{k+j+1}^{(i)} - Ay_1\| = \min_{\gamma}\|Bu_{k+j+1}^{(i)} - \gamma A(\mathbb{P}_p^{(i,l)})^{-1}(Bu_{k+j+1}^{(i)})\| \\ \leq\ & \|Bu_{k+j+1}^{(i)} - A(\mathbb{P}_p^{(i,l)})^{-1}(Bu_{k+j+1}^{(i)})\| \\ =\ & \|Bu_{k+j+1}^{(i)} - A(\mathbb{P}_p^{(i,l)})^{-1}(Bu_p c_p^{(i,l)} + Bu_p^{\perp} s_p^{(i,l)})\| \\ =\ & \|Bu_p c_p^{(i,l)} + Bu_p^{\perp} s_p^{(i,l)} - A(A^{-1}Bu_p)c_p^{(i,l)} - A(\mathbb{P}_p^{(i,l)})^{-1}Bu_p^{\perp} s_p^{(i,l)})\| \\ =\ & s_p^{(i,l)}\|(A(\mathbb{P}_p^{(i,l)})^{-1} - I)Bu_p^{\perp}\|. \end{aligned} \tag{5.15}$$

Therefore the relative residual norm is $s_p^{(i,l)}\frac{\|(A(\mathbb{P}_p^{(i,l)})^{-1}-I)Bu_p^{\perp}\|}{\|Bu_{k+j+1}^{(i)}\|} = O(s_p^{(i,l)})$.

Finally, Phase II of Algorithm 5.2 requires that

$$\frac{\|Bu_{k+j+1}^{(i)} - Ay_{q+1}\|}{\|Bu_{k+j+1}^{(i)}\|} = \frac{\|Bu_{k+j+1}^{(i)} - Ay_{q+1}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}\frac{\|Bu_{k+j+1}^{(i)} - Ay_1\|}{\|Bu_{k+j+1}^{(i)}\|} \leq \delta, \tag{5.16}$$

which is satisfied if and only if the relative residual of the correction equation

$$\frac{\|(Bu_{k+j+1}^{(i)} - Ay_1) - Az_q\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|} = \frac{\|Bu_{k+j+1}^{(i)} - Ay_{q+1}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|} \leq \frac{\delta\|Bu_{k+j+1}^{(i)}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|} = \frac{\delta}{O(s_p^{(i,l)})}. \tag{5.17}$$

The proof is thus concluded. ▮

Remark 3.3. The theorem shows that y_1 obtained in Phase I of Algorithm 5.2 equals $\mathcal{A}u_p c_p^{(i,l)}$ plus a small quantity proportional to $s_p^{(i,l)}$. As $u_p \in \mathcal{U}_p^{(i,l)}$, $\mathcal{A}u_p \in$

$\mathcal{A}\mathcal{U}_p^{(i,l)} = \mathcal{X}_p^{(i,l)}$; see (5.11). Recall that $X_p^{(i,l)}$ consists of the "solution vectors" of the linear systems in previous Arnoldi steps. Therefore, by constructing tuning as in Section 5.3.1 and applying one step of preconditioned GMRES with tuning to $Ay = Bu_{k+j+1}^{(i)}$, we get a good approximate solution y_1 which is roughly a linear combination of those solution vectors. The reason for the success of this approach is that $\angle(Bu_{k+j+1}^{(i)}, B\mathcal{U}_p^{(i,l)})$ is small, i.e., $Bu_{k+j+1}^{(i)}$ is roughly a linear combination of the right hand sides of the previously solved systems. This perspective is quite different from the motivation of tuning in all previous literature [25, 26, 65] and [28].

Remark 3.4. The theorem shows that a good approximate solution y_1 can be computed inexpensively in Phase I by tuning so that $\frac{\|Bu_{k+j+1}^{(i)}-Ay_1\|}{\|Bu_{k+j+1}^{(i)}\|} = O(s_p^{(i,l)}) \ll 1$. In fact, a valid y_1 can also be obtained in other ways, in particular, by solving a least squares problem

$$(5.18) \qquad \min_f \|Bu_{k+j+1}^{(i)} - AX_p^{(i,l)}f\|,$$

which can be easily done using the QR decomposition of $AX_p^{(i,l)} = BU_p^{(i,l)}$ (recall the definition of $X_p^{(i,l)}$ in (5.11)). Given that $u_{k+j+1}^{(i)} = u_p c_p^{(i,l)} + u_p^{\perp} s_p^{(i,l)}$, where $u_p \in \mathcal{U}_p^{(i,l)}$ and $u_p^{\perp} \perp \mathcal{U}_p^{(i,l)}$, we have

$$(5.19) \qquad \begin{aligned} \min_{f\in\mathbb{C}^p} \|Bu_{k+j+1}^{(i)} - AX_p^{(i,l)}f\| &= \min_{f\in\mathbb{C}^p} \|Bu_{k+j+1}^{(i)} - BU_p^{(i,l)}f\| \\ \leq\ & \|B(u_p c_p^{(i,l)} + u_p^{\perp} s_p^{(i,l)}) - Bu_p c_p^{(i,l)}\| = s_p^{(i,l)}\|Bu_p^{\perp}\|. \end{aligned}$$

Therefore, with $y_1 = X_p^{(i,l)}f$, we have $\frac{\|Bu_{k+j+1}^{(i)}-Ay_1\|}{\|Bu_{k+j+1}^{(i)}\|} = s_p^{(i,l)}\frac{\|Bu_p^{\perp}\|}{\|Bu_{k+j+1}^{(i)}\|} = O(s_p^{(i,l)})$. Similar to the observation in Chapter 4, the Phase I computation is somewhat cheaper for the least squares approach than the one step tuned preconditioned GMRES, but the former method required slightly more iterations in Phase II for our test problems, and the total inner iteration counts are about the same for the two approaches. In the following, for the sake of brevity, we only study the two-phase strategy where tuning is applied in Phase I.

Remark 3.5. Due to the one-step large reduction of the linear residual norm in Phase I, the stopping criterion in Algorithm 5.2 that $\frac{\|Bu^{(i)}_{k+j+1}-Ay_{q+1}\|}{\|Bu^{(i)}_{k+j+1}\|} \leq \delta$ is satisfied if and only if the relative residual of the correction equation $\frac{\|(Bu^{(i)}_{k+j+1}-Ay_1)-Az_q\|}{\|Bu^{(i)}_{k+j+1}-Ay_1\|}$ is bounded by the much less stringent relative tolerance $\frac{\delta\|Bu^{(i)}_{k+j+1}\|}{\|Bu^{(i)}_{k+j+1}-Ay_1\|} = \frac{\delta}{O(s_p^{(i,l)})} \gg \delta$. This larger relative tolerance implies that the inner iterations required for solving the correction equation can be considerably smaller than those needed to solve the original equation directly.

5.3.3 Linear solvers with subspace recycling for the correction equation

Step 2 of Algorithm 5.2 can be further refined by the use of linear solvers with subspace recycling to further reduce the number of inner iterations. This methodology has proved efficient for solving a long sequence of *slowly-changing* linear systems. When the iterative solution of one linear system is done, a small set of vectors from the current subspace for the candidate solutions is selected and "recycled", i.e., used for the solution of the next system in the sequence. Subspace recycling usually reduces the cost of solving subsequent linear systems, because the iterative solver does not have to build the candidate solution subspace from scratch. A popular solver of this type is the Generalized Conjugate Residual with implicit inner Orthogonalization and Deflated Restarting (GCRO-DR) [61] developed using ideas of special truncation [14] and restarting [55] for solving a single linear system.

Reference [61] makes a general assumption that the preconditioned system matrix changes from one linear system to the next, and thus the recycled subspace taken from the previous system must be transformed by matrix-vector products involving the current system matrix to fit into the solution of the current system. In the setting of solving the sequence of correction equations in Algorithm 5.2, fortunately, this transformation can be avoided, because the preconditioned system matrix without tuning is the same for the correction equation in all Arnoldi steps.

It is suggested in [61] that the harmonic Ritz vectors corresponding to smallest harmonic Ritz values can be chosen to span the recycled subspaces. These vectors are approximate eigenvectors of the preconditioned system matrix corresponding to smallest eigenvalues. If the harmonic Ritz vectors are good approximate eigenvectors, this strategy tends to reduce the duration of the initial latency of GMRES convergence typically observed when the system matrix has some eigenvalues of very small magnitude; see [20]. Our subspace recycling also includes dominant Ritz vectors, as suggested in [61]. In Section 5.5, our numerical experiments show that the set of dominant Ritz vectors is an effective choice for subspace recycling if the use of harmonic Ritz vectors fails to reduce the inner iteration counts.

5.4 A refined analysis of allowable errors in Arnoldi steps

Reference [8] is one of the earliest papers on inexact Krylov subspace eigenvalue algorithms, where a large number of numerical tests were carried out for the ordinary Arnoldi method (without restarting). It was observed empirically that the matrix-vector products involving $\mathcal{A}$ must be computed with high accuracy in the initial Arnoldi steps, whereas the accuracy can be relaxed as the Arnoldi method proceeds. A similar phenomenon is also observed in [35] for an inexact Lanczos method. An analysis based on matrix perturbation theory given in [74] shows that the allowable errors in the Arnoldi steps can be relaxed to a quantity inversely proportional to the eigenvalue residual norm of the current desired approximate invariant subspace, while the quality of the approximate invariant subspace is still under control (and is expected to improve) after these inexact Arnoldi steps. This *relaxation strategy* is extended in [28] to the inexact IRA method, where a practical estimate of the allowable tolerance is proposed for the linear systems in Arnoldi steps. Ideally, accurately estimated allowable tolerances can help reduce the inner iteration counts to the best extent possible without compromising the convergence of the IRA method to the desired invariant subspace. In this section, we give a refined analysis of allowable errors in Arnoldi steps and an alternative estimate of allowable tolerances for the linear systems.

Suppose the matrix-vector product involving $\mathcal{A} = A^{-1}B$ is applied inexactly for m Arnoldi steps, with an error f_j $(1 \leq j \leq m)$ introduced at each step. Thus we have the following inexact Arnoldi decomposition:

$$\mathcal{A}U_m + F_m = (\mathcal{A} + F_m U_m^*)U_m = U_m H_m + h_{m+1,m} u_{m+1} e_m^T, \tag{5.20}$$

where U_m spans a Krylov subspace of the perturbed matrix $\mathcal{A} + F_m U_m^*$. Let the Schur decomposition of H_m be

$$H_m = W_m T_m W_m^*, \text{ with } W_m = \begin{bmatrix} W_m^{11} & W_m^{12} \\ W_m^{21} & W_m^{22} \end{bmatrix} \text{ and } T_m = \begin{bmatrix} T_m^{11} & T_m^{12} \\ 0 & T_m^{22} \end{bmatrix}, \tag{5.21}$$

where $T_m^{11} \in \mathbb{C}^{k \times k}$, $T_m^{22} \in \mathbb{C}^{p \times p}$ (the size of other blocks can be determined accordingly). Assume that $\lambda(T_m^{11})$ are the wanted Ritz values, and $\lambda(T_m^{22})$ are the unwanted Ritz values. Then we use the Rayleigh-Ritz method (Section 4.1, Chapter 4 of [86]) to extract the desired approximate invariant subspace $U_m W_m^1$, where $W_m^1 = \begin{bmatrix} W_m^{11} \\ W_m^{21} \end{bmatrix}$ contains the wanted Ritz vectors. From (5.20) and (5.21), the corresponding eigenvalue residual is

$$\begin{aligned} \mathcal{A}U_m W_m^1 - U_m H_m W_m^1 &= \mathcal{A}U_m W_m^1 - U_m W_m^1 T_m^{11} \\ &= h_{m+1,m} u_{m+1} e_m^T W_m^1 - F_m W_m^1, \end{aligned} \tag{5.22}$$

from which follows

$$\|(\mathcal{A}U_m W_m^1 - U_m W_m^1 T_m^{11}) - h_{m+1,m} u_{m+1} e_m^T\| = \|F_m W_m^1\|. \tag{5.23}$$

Here, as introduced in Section 5.2, $\mathcal{A}U_mW_m^1 - U_mW_m^1T_m^{11}$ is the true eigenvalue residual, and $R_m = h_{m+1,m}u_{m+1}e_m^TW_m^1$ is the estimated residual (referred to as the "computed residual" in [74, 28]). The difference between the two residuals depends on $\|F_mW_m^1\|$. For the inexact Arnoldi method, we want to keep the quality of $U_mW_m^1$ under control in spite of the presence of the error matrix F_m. To achieve this goal, we need to control $\|F_mW_m^1\|$, such that the desired approximate invariant subspace $U_mW_m^1$ contained in U_m is not obviously contaminated by F_m, i.e., the true residual is still reasonably close to the estimated residual. This perspective addresses the concern in [89] that the estimated residual may be an unreliable estimate of the true residual, if the matrix-vector products in Arnoldi steps are computed with errors.

To see why the allowable errors at some Arnoldi steps can be relaxed, note that

$$\|F_mW_m^1\| \le \|F_kW_m^{11}\| + \|F_{k+1:m}W_m^{21}\| \le \|F_k\| + \|F_{k+1:m}\|\|W_m^{21}\|. \tag{5.24}$$

Therefore, for a given k-step inexact Arnoldi decomposition with a small enough $\|F_k\|$, $\|F_{k+1:m}\|$ does not have to be very small as long as $\|W_m^{21}\|$, i.e., the magnitude of the last $m-k$ entries of the wanted Ritz vectors W_m^1 (see (5.21)), is small enough. The next theorem, which extends Theorem 3.2 of [28], shows that $\|W_m^{21}\|$ is proportional to the estimated residual at step k.

Theorem 5.4.1 *Let $\mathcal{A}U_k + F_k = U_kH_k + h_{k+1,k}u_{k+1}e_k^T$ be a k-step inexact Arnoldi decomposition, where the Schur decomposition of H_k is $H_k = W_kT_kW_k^*$. We then carry out $m-k$ additional inexact Arnoldi steps to get an m-step decomposition $\mathcal{A}U_m + F_m = U_mH_m + h_{m+1,m}u_{m+1}e_m^T$. The Schur decomposition of H_m is given in* (5.21)*. Let $R_k = \mathcal{A}U_kW_k - U_kH_kW_k = h_{k+1,k}u_{k+1}e_k^TW_k$ be the estimated residual at Arnoldi step k. Then*

$$\frac{\|R_k\|}{\|R_k\| + \|\mathcal{S}_m\|} \le \|W_m^{21}\| \le \frac{\|R_k\|}{\operatorname{sep}(T_k, T_m^{22})}, \tag{5.25}$$

where $\mathcal{S}_m$ *is the Sylvester operator* $G \to \mathcal{S}_m(G) : T_m^{22}G - GT_k$, $\|\mathcal{S}_m\| = \max_{\|G\|=1} \|\mathcal{S}_m(G)\|$, *and* $\mathrm{sep}(T_k, T_m^{22}) = \min_{\|G\|=1} \|\mathcal{S}_m(G)\|$.

Proof We only need to prove the lower bound, as the upper bound is established in Theorem 3.2 of [28]. The estimated residual norm at step k is

$$(5.26) \qquad \|R_k\| = \|h_{k+1,k}u_{k+1}e_k^T W_k\| = h_{k+1,k}\|e_k^T W_k\| = h_{k+1,k}.$$

An interesting observation shows that the following quantity also equals the estimated residual norm:

$$(5.27) \qquad \left\| H_m \begin{pmatrix} W_k \\ 0 \end{pmatrix} - \begin{pmatrix} W_k \\ 0 \end{pmatrix} T_k \right\|$$

$$\begin{aligned}
&= \left\| \begin{pmatrix} H_k & H_m^{12} \\ h_{k+1,k}e_1e_k^T & H_m^{22} \end{pmatrix} \begin{pmatrix} W_k \\ 0 \end{pmatrix} - \begin{pmatrix} W_k \\ 0 \end{pmatrix} T_k \right\| \\
&= \left\| \begin{pmatrix} H_k \\ h_{k+1,k}e_1e_k^T \end{pmatrix} W_k - \begin{pmatrix} W_k \\ 0 \end{pmatrix} T_k \right\| = \left\| \begin{pmatrix} H_kW_k - W_kT_k \\ h_{k+1,k}e_1e_k^T W_k \end{pmatrix} \right\| \\
&= \left\| \begin{pmatrix} 0 \\ h_{k+1,k}e_1e_k^T W_k \end{pmatrix} \right\| = h_{k+1,k}\|e_k^T W_k\| = h_{k+1,k}.
\end{aligned}$$

Since W_m is unitary, with the first expression in the last line of (5.27), we have

$$(5.28) \|R_k\| = \left\| W_m^* \Big(H_m \begin{pmatrix} W_k \\ 0 \end{pmatrix} - \begin{pmatrix} W_k \\ 0 \end{pmatrix} T_k \Big) \right\|$$

$$= \left\| \begin{pmatrix} (W_m^{11})^* & (W_m^{21})^* \\ (W_m^{12})^* & (W_m^{22})^* \end{pmatrix} \begin{pmatrix} 0 \\ h_{k+1,k}e_1e_k^T W_k \end{pmatrix} \right\| = \left\| \begin{pmatrix} h_{k+1,k}(W_m^{21})^* e_1e_k^T W_k \\ h_{k+1,k}(W_m^{22})^* e_1e_k^T W_k \end{pmatrix} \right\|.$$

On the other hand, using the Schur decomposition of H_m, we also have

$$
\begin{aligned}
\|R_k\| &= \left\| W_m^* \left(H_m \begin{pmatrix} W_k \\ 0 \end{pmatrix} - \begin{pmatrix} W_k \\ 0 \end{pmatrix} T_k \right) \right\| \\
&= \left\| T_m W_m^* \begin{pmatrix} W_k \\ 0 \end{pmatrix} - W_m^* \begin{pmatrix} W_k \\ 0 \end{pmatrix} T_k \right\| \\
&= \left\| \begin{pmatrix} T_m^{11} & T_m^{12} \\ 0 & T_m^{22} \end{pmatrix} \begin{pmatrix} (W_m^{11})^* W_k \\ (W_m^{12})^* W_k \end{pmatrix} - \begin{pmatrix} (W_m^{11})^* W_k \\ (W_m^{12})^* W_k \end{pmatrix} T_k \right\| \\
&= \left\| \begin{pmatrix} T_m^{11}(W_m^{11})^* W_k - (W_m^{11})^* W_k T_k + T_m^{12}(W_m^{12})^* W_k \\ T_m^{22}(W_m^{12})^* W_k - (W_m^{12})^* W_k T_k \end{pmatrix} \right\|.
\end{aligned} \tag{5.29}
$$

Since all matrices except for R_k in the argument of the norm operator in (5.28) and (5.29) are identical, we take the upper block from (5.28) and lower block from (5.29) to get

$$
\begin{aligned}
\|R_k\| &= \left\| \begin{pmatrix} h_{k+1,k}(W_m^{21})^* e_1 e_k^T W_k \\ T_m^{22}(W_m^{12})^* W_k - (W_m^{12})^* W_k T_k \end{pmatrix} \right\| \\
&\leq \|h_{k+1,k}(W_m^{21})^* e_1 e_k^T W_k\| + \|T_m^{22}(W_m^{12})^* W_k - (W_m^{12})^* W_k T_k\| \\
&\leq h_{k+1,k}\|e_k^T W_k\| \|e_1^T W_m^{21}\| + \|\mathcal{S}_m\| \|(W_m^{12})^* W_k\| \\
&= \|R_k\| \|e_1^T W_m^{21}\| + \|\mathcal{S}_m\| \|W_m^{12}\| \leq \|R_k\| \|W_m^{21}\| + \|\mathcal{S}_m\| \|W_m^{21}\|.
\end{aligned} \tag{5.30}
$$

Note that in the last line of (5.30), $\|(W_m^{12})^* W_k\| = \|W_m^{12}\| = \|W_m^{21}\|$ (Theorem 2.6.1 in Golub and van Loan [32]). The lower bound in (5.25) is thus established. $\blacksquare$

As observed above, for a given k-step inexact Arnoldi decomposition with small $\|F_k\|$, the error matrix $\|F_{k+1:m}\|$ associated with the upcoming $m-k$ inexact Arnoldi steps must be controlled appropriately to make sure that $U_m W_m^1$ is not obviously contaminated after these steps. In particular, $\|f_{k+1}\|$ cannot be too big. The following theorem gives an upper bound of $\|f_{k+1}\|$.

Theorem 5.4.2 *Given $\epsilon_1 > 0$, suppose we have a k-step inexact Arnoldi decomposition $\mathcal{A}U_k + F_k = U_k H_k + h_{k+1,k} u_{k+1} e_k^T$, where $\|F_k\| \le \epsilon_1$. Then for the next Arnoldi step,*

$$\|f_{k+1}\| \le \left(1 + \frac{\|\mathcal{S}_{k+1}\|}{\|R_k\|}\right)(\epsilon_1 + \epsilon_2) \quad \text{(where } \mathcal{S}_{k+1} \text{ is defined in Theorem 5.4.1)} \tag{5.31}$$

is a necessary condition to make $\|(\mathcal{A}U_{k+1}W_{k+1}^1 - U_{k+1}W_{k+1}^1 T_{k+1}^{11}) - R_{k+1}\| \le \epsilon_2$

Proof Let $m = k+1$. We have the following estimate of the difference between the computed and true residual,

$$\begin{aligned}
(5.32) \qquad & \|(\mathcal{A}U_{k+1}W_{k+1}^1 - U_{k+1}W_{k+1}^1 T_{k+1}^{11}) - R_{k+1}\| = \|F_{k+1}W_{k+1}^1\| \\
= \quad & \|F_k W_{k+1}^{11} + f_{k+1} W_{k+1}^{21}\| \ge \|f_{k+1}W_{k+1}^{21}\| - \|F_k W_{k+1}^{11}\| \\
\ge \quad & \|f_{k+1}\|\|W_{k+1}^{21}\| - \|F_k\|\|W_{k+1}^{11}\| \ge \|f_{k+1}\| \frac{\|R_k\|}{\|R_k\| + \|\mathcal{S}_{k+1}\|} - \epsilon_1.
\end{aligned}$$

Note that $\|f_{k+1}W_{k+1}^{21}\| = \|f_{k+1}\|\|W_{k+1}^{21}\|$ because f_{k+1} and W_{k+1}^{21} are respectively a column vector and row vector, and $\|W_{k+1}^{11}\| \le \|W_{k+1}^1\| = 1$. It follows immediately that (5.32) is bigger than ϵ_2 if $\|f_{k+1}\| > (1 + \frac{\|\mathcal{S}_{k+1}\|}{\|R_k\|})(\epsilon_1 + \epsilon_2)$. ∎

In practice, we usually choose $\epsilon_2 = \epsilon_1$ and denote both quantities as ϵ. Similarly, using the upper bound of $\|W_m^{21}\|$ in Theorem 5.4.1, we can show that $\|f_{k+1}\| \le \frac{\text{sep}(T_m^{22}, T_k)}{\|R_k\|}\epsilon$ is sufficient to make $\|(\mathcal{A}U_{k+1}W_{k+1}^1 - U_{k+1}W_{k+1}^1 T_{k+1}^{11}) - R_{k+1}\| \le 2\epsilon$. However, both conditions for $\|f_{k+1}\|$ might not be appropriate to estimate the actual allowable error in the (k+1)-th Arnoldi step. In fact, $\text{sep}(T_{k+1}^{22}, T_k)$ and $\|\mathcal{S}_{k+1}\|$ are analogous to the smallest and the largest singular values of the Sylvester operator $\mathcal{S}_{k+1}$. The necessary condition is generally too weak, as an obviously smaller $\|f_{k+1}\|$ may still not suffice to keep the approximate invariant subspace from being contaminated. On the other hand, the sufficient condition might be overly conservative, giving excessively small tolerance for the linear system $Ay = Bu_{k+j+1}$ $(0 \le j \le m-k-1)$ and leading to unnecessary extra inner iterations. To give a practical estimate of the allowable

$\|F_{k+1:m}\|$, [28] substitutes $\min|\lambda(T_k)-\lambda(T_m^{22})|$ for $\mathrm{sep}(T_m^{22},T_k)$, which is difficult to estimate. Since $\min|\lambda(T_k)-\lambda(T_m^{22})| > \mathrm{sep}(T_m^{22},T_k)$ for non-normal $\mathcal{A}$, this substitution essentially gives a less conservative estimate.

A better estimate should be a trade-off between the two conditions. Theorem 3.2 of [28] uses $\|T_m^{22}(W_m^{12})^* W_k - (W_m^{12})^* W_k T_k\| \geq \mathrm{sep}(T_m^{22},T_k)\|(W_m^{12})^* W_k\|$, whereas Theorem 5.4.1 above applies $\|T_m^{22}(W_m^{12})^* W_k - (W_m^{12})^* W_k T_k\| \leq \|\mathcal{S}_m\|\|(W_m^{12})^* W_k\|$. Therefore, a more accurate estimate can be obtained by replacing the lower bound $\mathrm{sep}(T_m^{22},T_k)$ and upper bound $\|\mathcal{S}_m\|$ by

$$\frac{\|T_m^{22}(W_m^{12})^* W_k - (W_m^{12})^* W_k T_k\|}{\|(W_m^{12})^* W_k\|} = \frac{\|T_m^{22}(W_m^{12})^* - (W_m^{12})^* H_k\|}{\|W_m^{12}\|}, \tag{5.33}$$

which takes into account the actual effect of $\mathcal{S}_m$ on $(W_m^{12})^* W_k$. Here we use the fact that $H_k = W_k T_k W_k^*$ is a Schur decomposition.

The above strategy gives a theoretically more accurate estimate of $\|F_{k+1:m}\|$. However, like the estimate $\min|\lambda(T_k)-\lambda(T_m^{22})|$ in [28], it depends on the Schur decomposition of H_m, which is not available at step k. The practical (heuristic) solution is to use the decomposition of H_m from the previous IRA cycle and H_k of the current cycle. Specifically, suppose at the beginning of the ith IRA cycle, we have $\mathcal{A}U_k^{(i)} + F_k^{(i)} = U_k^{(i)} H_k^{(i)} + h_{k+1,k}^{(i)} u_{k+1}^{(i)} e_k^T$. Then we define

$$\sigma_{est}^{(i)} \equiv \frac{\|T_m^{22(i-1)}(W_m^{12(i-1)})^* - (W_m^{12(i-1)})^* H_k^{(i)}\|}{\|W_m^{12(i-1)}\|}, \tag{5.34}$$

which is very easy to compute. Note that $H_k^{(i)} = \tilde{H}_k^{(i-1)}$ if the exact shift strategy is used; see (5.3). Substituting $\sigma_{est}^{(i)}$ for $\mathrm{sep}(T_m^{22(i)}, T_k^{(i)})$ in the relaxation strategy (3.13) in [28], we have the following heuristic estimate of the allowable errors:

$$\begin{aligned} \|f_j^{(i)}\| &\leq \frac{\epsilon}{2k} && (i=1, 1\leq j\leq m), \quad \text{and} \\ \|f_{k+j+1}^{(i)}\| &\leq \frac{\epsilon}{2(m-k)}\frac{\sigma_{est}^{(i)}}{\|R_k^{(i)}\|} && (i>1, 0\leq j\leq m-k-1). \end{aligned} \tag{5.35}$$

Remark. To the best of our knowledge, given a k-step inexact Arnoldi decomposition with small $\|F_k\|$, none of the existing *practical* (computable) estimates of allowable $\|F_{k+1:m}\|$ can theoretically guarantee that the desired approximate invariant subspace $U_m W_m^1$ will not be contaminated after $m-k$ inexact Arnoldi steps. The estimate in [74] is for the ordinary Arnoldi method, assuming that the computed eigenvalue residual at step k is already small enough, which might not be the case for a given k-step Arnoldi decomposition; reference [28] uses $\min|\lambda(T_k^{(i)}) - \lambda(T_m^{22(i-1)})|$ in place of $\text{sep}(T_k^{(i)}, T_m^{22(i)})$, which is replaced by $\sigma_{est}^{(i)}$ in our new estimate. We will compare the new estimate with that in [28] in Section 5.5.

Finally, we point out the $\|F_m\|$ should be properly scaled. In fact, as $\mathcal{A}U_m + F_m = U_m H_m + h_{m+1,m} u_{m+1} e_m^T$, the *relative* quantity $\frac{\|F_m\|}{\|\mathcal{A}U_m\|}$ should be used to measure the magnitude of errors, especially if $\|\mathcal{A}U_m\|$ is too small or too large. Specifically, at the $(k+j+1)$-th Arnoldi step, the linear system $Ay = Bu_{k+j+1}$ needs to be solved inexactly. The relative error $\frac{\|f_{k+j+1}\|}{\|\mathcal{A}u_{k+j+1}\|} = \frac{\|y - A^{-1}Bu_{k+j+1}\|}{\|A^{-1}Bu_{k+j+1}\|}$ is not available because we do not have $A^{-1}Bu_{k+j+1}$. A reasonable and convenient substitute is the relative residual norm of this linear system $\frac{\|Ay - Bu_{k+j+1}\|}{\|Bu_{k+j+1}\|}$. For our inexact IRA method, we require this quantity to be bounded above by the new estimate in (5.35).

5.5 Numerical Experiments

We present and discuss the results of numerical experiments in this section, showing the effectiveness of the new tuning strategy, subspace recycling and the new estimated relaxation. The following issues are addressed:

1. We show that the tuning strategy constructed using solution vectors obtained from previous Arnoldi steps works as Theorem 5.3.3 describes: a single GMRES step with this tuning applied to $Ay = Bu_{k+j+1}^{(i)}$ gives a good approximate solution y_1 for which the residual norm $\frac{\|Bu_{k+j+1}^{(i)} - Ay_1\|}{\|Bu_{k+j+1}^{(i)}\|} = O(s_p^{(i,l)}) \ll 1$, and therefore the correction equation can be solved with a less stringent relative tolerance

$\frac{\delta\|Bu_{k+j+1}^{(i)}\|}{\|Bu_{k+j+1}^{(i)}-Ay_1\|} \gg \delta$, no matter if δ is a fixed or relaxed relative tolerance for $Ay = Bu_{k+j+1}^{(i)}$. The new tuning strategy is compared with the original tuning strategy in [28].

2. We compare inexact IRA methods with non-relaxed (fixed) tolerances $\delta_f = \frac{\epsilon}{2k}$ where ϵ and k are given in Table 5.1 (this δ_f is used in [28] for inexact IRA with a fixed tolerance for the inner solve) and relaxed tolerances δ_r given by either the original estimate in [28] or the new estimate in (5.35). The accuracy of the two estimates is discussed based on the numerical results.

3. We show that further reduction of inner iteration counts can be achieved at little cost by proper subspace recycling.

We first explain the stopping criterion for the inexact IRA method. Suppose at the beginning of the ith IRA cycle, we have $\mathcal{A}U_k^{(i)} + F_k^{(i)} = U_k^{(i)}H_k^{(i)} + h_{k+1,k}^{(i)}u_{k+1}^{(i)}e_k^T$. Let $(\theta_j^{(i)}, v_j^{(i)})$ $(1 \leq j \leq k)$ be a Ritz pair, i.e., an eigenpair of $H_k^{(i)}$. Post-multiplying the above equation by $v_j^{(i)}$, we have

$$\mathcal{A}(U_k^{(i)}v_j^{(i)}) - \theta_j^{(i)}(U_k^{(i)}v_j^{(i)}) - (h_{k+1,k}^{(i)}v_{kj}^{(i)})u_{k+1}^{(i)} = -F_k^{(i)}v_j^{(i)}, \tag{5.36}$$

where $v_{kj}^{(i)}$ is the k-th (last) entry of $v_j^{(i)}$. Here $U_k^{(i)}v_j^{(i)}$ is an approximate eigenvector of $\mathcal{A}$, $\mathcal{A}(U_k^{(i)}v_j^{(i)}) - \theta_j^{(i)}(U_k^{(i)}v_j^{(i)})$ is the true eigenvalue residual, and $(h_{k+1,k}^{(i)}v_{kj}^{(i)})u_{k+1}^{(i)}$ is the estimated residual. As the magnitude of errors has been kept under control to guarantee that the true residual is close enough to the estimated one, we check if

$$\left|\frac{h_{k+1,k}^{(i)}v_{kj}^{(i)}}{\theta_j^{(i)}}\right| \approx \frac{\|\mathcal{A}(U_k^{(i)}v_j^{(i)}) - \theta_j^{(i)}(U_k^{(i)}v_j^{(i)})\|}{|\theta_j^{(i)}|} \tag{5.37}$$

is smaller than some prescribed tolerance ϵ_{eig}. Using estimated residuals avoids the overhead of the Rayleigh-Ritz procedure.

Another issue is that k does not have to be equal to the number of desired eigenpairs k_w. One can choose a slightly bigger k for the IRA method, and only test $|(\theta_j^{(i)})^{-1}h_{k+1,k}^{(i)}v_{kj}^{(i)}|$ in (5.37) for $1 \le j \le k_w$. Our experience is that for fixed $m{-}k$ (the number of Arnoldi steps in each *restarted* IRA cycle), more often than not, this choice of k reduces the number of IRA cycles. We speculate that this strategy makes the unwanted Ritz values more separated from the desired eigenvalues; therefore it is less likely for the filter polynomial to damp the desired eigenvector components during the restart.

Four test problems are used in our numerical experiments. The first is SHERMAN5 from MatrixMarket [51], a real matrix of order 3312 arising from oil reservoir modeling. We use the shift-invert operator $\mathcal{A} = A^{-1}$ (with $B = I$) to detect some eigenvalues closest to zero. The inner solve is done with ILU preconditioner with drop tolerance 0.008 given by MATLAB's `ilu` function. This example is used in [28] to show the effectiveness of the tuning and the relaxation strategy therein.

The second through the fourth test problems are used in Chapter 4 for inexact subspace iteration. The parameters used to solve the test problems are given below; their values for each individual problem are summarized in Table 5.1.

1. k_w, k, m – we use the IRA method to compute k_w eigenpairs; m and k are the order of the Arnoldi decomposition right before and after the implicit restart.

2. $\sigma, \sigma_1, \sigma_2$ – the shifts of $\mathcal{A} = (A - \sigma B)^{-1}B$ and $\mathcal{A} = (A - \sigma_1 B)^{-1}(A - \sigma_2 B)$

3. τ – we stop the IRA method if the estimated residual in (5.37) is smaller than τ for all k_w desired approximate eigenpairs

4. ϵ – the small quantity used in 5.35) to estimate the allowable tolerances for the linear systems

5. $l_1, l_2 - l_1$ harmonic Ritz vectors corresponding to harmonic Ritz values of smallest magnitude and l_2 dominant Ritz vectors are used for subspace recycling

	k_w	p	k	σ (σ_1)	σ_2	τ	ϵ	l_1	l_2
Prob 1	8	8	12	0	–	2×10^{-14}	2×10^{-11}	10	10
Prob 2	3	4	9	-0.0325	0.125	2×10^{-9}	2×10^{-10}	10	10
Prob 3(a)	5	7	13	0	–	5×10^{-10}	5×10^{-9}	0	30
Prob 3(b)					-0.46				
Prob 4(a)					–			0	45
Prob 4(b)					-0.24				

Table 5.1: Parameters used to solve the test problems

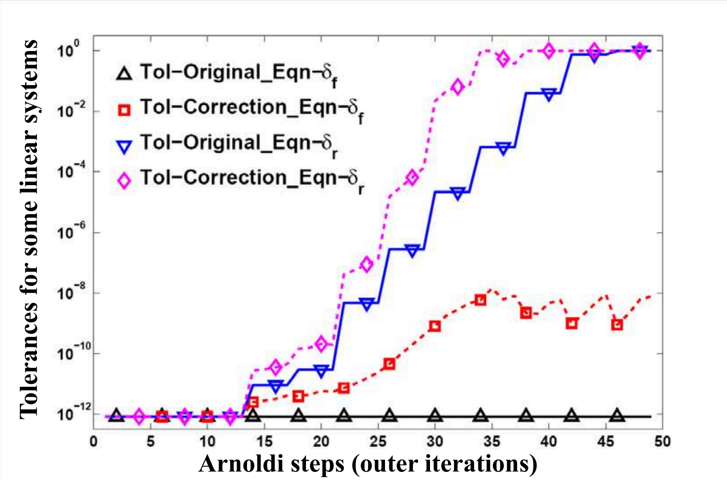

Figure 5.1: Problem 1: relative tolerances for the original systems and correction equations

Figure 5.1 plots the relative tolerances δ for the original systems $Ay = Bu^{(i)}_{k+j+1}$ (solid lines) and the derived relative tolerances $\frac{\delta\|Bu^{(i)}_{k+j+1}\|}{\|Bu^{(i)}_{k+j+1}-Ay_1\|}$ for the correction equations $Az = Bu^{(i)}_{k+j+1} - Ay_1$ (dashed lines) against the Arnoldi steps for Problem 1. It corroborates the property of the two-phase algorithm described in Theorem 5.3.3. Specifically, by applying one step of GMRES with the new tuning to the original system, we get a good approximate solution y_1 for which the relative residual norm $\frac{\|Bu^{(i)}_{k+j+1}-Ay_1\|}{\|Bu^{(i)}_{k+j+1}\|} = O(s_p^{(i,l)}) \ll 1$, and therefore the derived relative tolerance of the correction equation $\frac{\delta\|Bu^{(i)}_{k+j+1}\|}{\|Bu^{(i)}_{k+j+1}-Ay_1\|} \gg \delta$. The curves in Figure 5.1 are as follows:

- △ Tol-Original_Eqn-δ_f and □ Tol-Correction_Eqn-δ_f – The fixed relative tolerance $\delta_f = \frac{\epsilon}{2k} \approx 10^{-12}$ for the original system $Ay = Bu^{(i)}_{k+j+1}$ and the derived relative tolerance $\frac{\delta_f \|Bu_{k+j+1}^{(i)}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}$ for the correction equation $Az = Bu_{k+j+1}^{(i)} - Ay_1$.

- $\triangledown$ Tol-Original_Eqn-δ_r and $\diamondsuit$ Tol-Correction_Eqn-δ_r – The relaxed relative tolerances δ_r estimated by (5.35) for $Ay = Bu_{k+j+1}^{(i)}$ and the derived relative tolerance $\frac{\delta_r \|Bu_{k+j+1}^{(i)}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}$ for $Az = Bu_{k+j+1}^{(i)} - Ay_1$.

In the tests in Figure 5.1, Phase I computations use the new tuning strategy with solution vectors from the current and 5 previous IRA cycles. We see from Figure 5.1 that the tolerances $\frac{\delta \|Bu_{k+j+1}^{(i)}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}$ for the correction equations (dashed lines) are obviously larger than the tolerances δ for the original systems (solid lines), no matter if δ stands for the fixed tolerance δ_f ($\triangle$) or relaxed ($\triangledown$) tolerance δ_r. For example, suppose fixed tolerances are used for the original systems. In the 34th Arnoldi step, the derived tolerance $\frac{\delta_f \|Bu_{k+j+1}^{(i)}\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}$ for the correction equation $Az = Bu_{k+j+1}^{(i)} - Ay_1$ is about 10^{-8}, or 10^4 times as large as the tolerance δ_f for the original system $Ay = Bu_{k+j+1}^{(i)}$. The larger tolerances indicate that fewer inner iterations are needed to solve the correction equations than those required to solve the original systems without tuning.

The reduction of inner iterations by the two-phase algorithm (with the new tuning used in Phase I) can be seen from Figure 5.2, where the inner iteration counts required by three different strategies for solving $Ay = Bu_{k+j+1}^{(i)}$ are plotted against the Arnoldi steps:

- "No Tuning" (dotted line) – Solve the original systems $Ay = Bu_{k+j+1}^{(i)}$ by preconditioned GMRES to the fixed tolerance $\delta_f = \frac{\epsilon}{2k}$ without any enhancements.

- "Original Tuning" ($\triangle$, solid line) – Solve $Ay = Bu_{k+j+1}^{(i)}$ by GMRES with the original version of tuning in [28] to the fixed tolerance δ_f (note that the original tuning needs to be applied at each GMRES step; it cannot work with the two-phase strategy).

- "New Tuning (5 previous cycles)" ($\square$, dashed line) – Solve $Ay = Bu_{k+j+1}^{(i)}$ by the

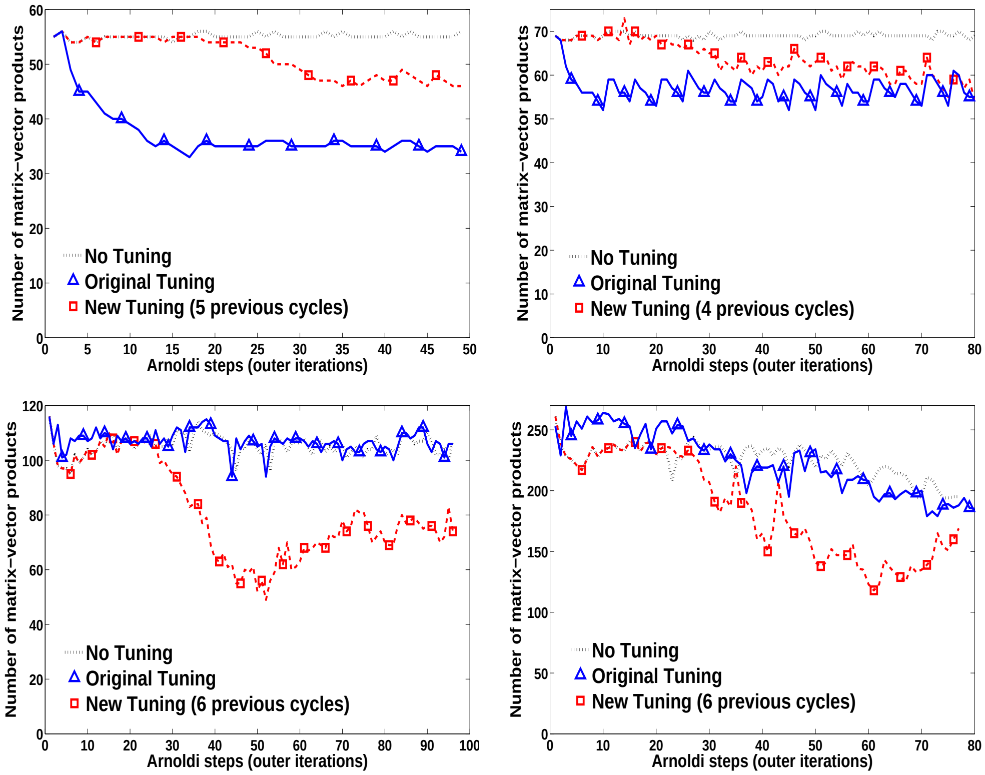

Figure 5.2: Performance of different strategies with fixed tolerances of inner solves for Problems 1, 2, 3(a) and 4(a)

two-phase algorithm to the fixed tolerance δ_f; in the first phase, the new tuning is constructed using solution vectors from the current and 5 previous IRA cycles.

Clearly, compared to the "No Tuning" strategy, the two-phase algorithm (□, with the new tuning used in Phase I) requires fewer inner iterations due to the larger relative tolerances for the correction equations.

We now compare the effectiveness of the two-phase algorithm (the new tuning is used in Phase I only for a single GMRES step) and the original tuning strategy that is applied at each GMRES step. In Figure 5.2, we see that for Problems 1 and 2, the relative reduction of inner iterations obtained by the use of the two-phase algorithm is only 15% at most; for Problems 3(a) and 4(a), however, the relative reduction can be as large as $30\% - 40\%$. On the other hand, the original tuning strategy works well for the first two problems, but fails to obviously reduce the inner iterations for the latter two.

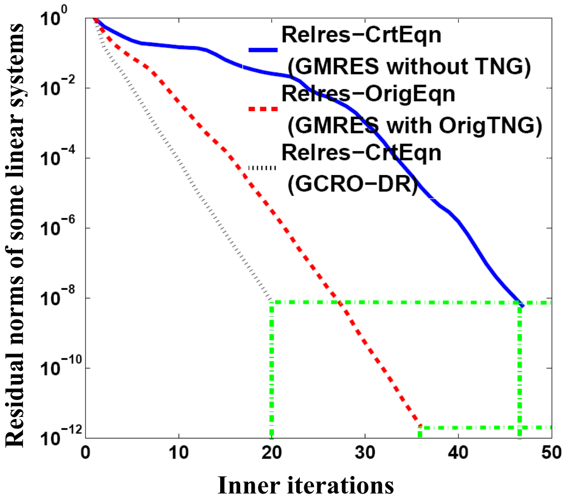

Figure 5.3: Problem 1: residual norms of $Ay = Bu_j^{(i)}$ or $Az = Bu_j^{(i)} - Ay_1$ at a single Arnoldi step, for three solution strategies

The reason can be seen from Figure 5.3, which plots the relative residual norms of $Ay = Bu_{k+j+1}^{(i)}$ or $Az = Bu_{k+j+1}^{(i)} - Ay_1$ against inner iterations at a single Arnoldi step (step 34):

- Relres-CrtEqn (Solid line) – relative residual norms $\frac{\|(Bu_{k+j+1}^{(i)} - Ay_1) - Az_q\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}$ of the correction equation $Az = Bu_{k+j+1}^{(i)} - Ay_1$ solved by preconditioned GMRES without tuning in Phase II of the two-phase strategy.

- Relres-OrigEqn (Dashed line) – residual norms $\frac{\|Bu_{k+j+1}^{(i)} - Ay_q\|}{\|Bu_{k+j+1}^{(i)}\|}$ of the original system $Ay = Bu_{k+j+1}^{(i)}$ solved by GMRES with the original version of tuning (two-phase strategy cannot be used here).

- Relres-CrtEqn (Dash-dot line) – residual norms $\frac{\|(Bu_{k+j+1}^{(i)} - Ay_1) - Az_q\|}{\|Bu_{k+j+1}^{(i)} - Ay_1\|}$ of the correction equation solved by GCRO-DR (using subspace recycling) in Phase II of the two-phase strategy.

Here a fixed tolerance $\delta_f = \frac{\epsilon}{2k}$ is used for $Ay = Bu_{k+j+1}^{(i)}$. We include the behavior of the linear solver with subspace recycling (GCRO-DR), for reasons explained below. In this Arnoldi step, we see from Figure 5.1 that the relative tolerances for $Ay = Bu_{k+j+1}^{(i)}$

and $Az = Bu^{(i)}_{k+j+1} - Ay_1$ are $\delta_f \approx 10^{-12}$ and $\frac{\delta_f \|Bu^{(i)}_{k+j+1}\|}{\|Bu^{(i)}_{k+j+1} - Ay_1\|} \approx 10^{-8}$ respectively. Therefore, as shown in Figure 5.3, 47 GMRES steps without tuning are needed to solve the correction equation (48 GMRES steps in total for solving the original system, including the one step in Phase I), 36 GMRES steps with the original tuning are needed for solving the original system, and only 20 GCRO-DR stepsare required for the correction equation (21 inner steps in total for solving the original system). The asymptotic convergence rate of the three solves are roughly the same.

It is clear from Figure 5.3 that there are two types of strategy to reduce the inner iteration counts: one is to reduce the length of the latencies observed in the initial inner iteration steps, and the other is to use larger tolerances for the inner solves. The first type of strategy include the original tuning and linear solvers with subspace recycling. Specifically, let P be an existing untuned preconditioner, and $\mathbb{P}$ be the tuned version defined in [28]. It is shown that the preconditioned operator $A\mathbb{P}^{-1}$ tends to have better eigenvalue clustering than AP^{-1}, especially if P is not strong. For linear solvers with subspace recycling, the recycled subspaces are spanned by approximate eigenvectors corresponding to smallest and/or largest eigenvalues of AP^{-1}. Both approaches essentially eliminate the eigenvalues of smallest and/or largest magnitude of the preconditioned system matrix; these are usually the source of the initial latencies exhibited during the inner iterations.

However, the original tuning is *not* always effective for this purpose. As Figure 5.2 shows, for Problems 3 and 4, the solution strategy with the original tuning (△) requires almost as many inner iterations as the plain inexact IRA (dotted line), while the new tuning (□) reduces the inner iteration counts considerably. The reason is that the linear solves for these two problems are performed with the least square commutator preconditioner [19, 20], for which the preconditioned system matrix AP^{-1} has most eigenvalues clustered around 1 and only a small number of outliers [20]. For this strong preconditioner, in our experience, it is hard for the original tuning to further cluster the eigenvalues and reduce the initial latencies of GMRES iterations.

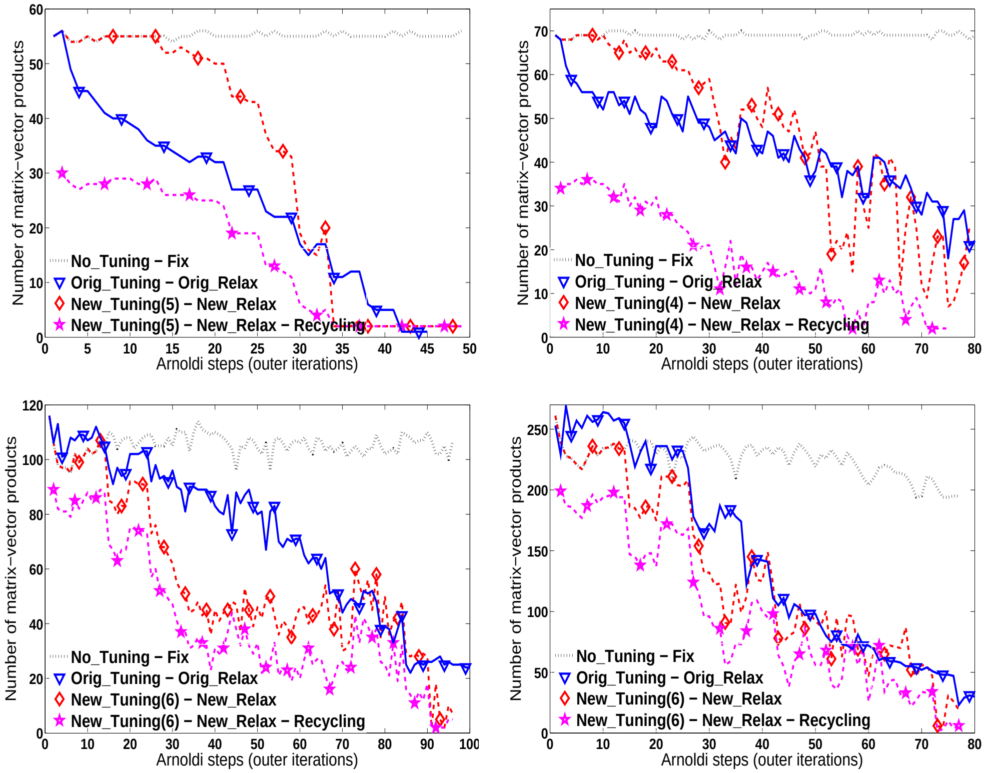

Figure 5.4: Performance of different strategies with relaxed tolerances of inner solves for Problems 1, 2, 3(a) and 4(a)

The second type of strategy includes the two-phase algorithm and the relaxation strategy. The motivation of the two-phase algorithm is to generate a good approximate solution y_1 for $Ay = Bu^{(i)}_{k+j+1}$ by one GMRES step with the new tuning, such that the tolerance for the correction equation can be much larger than that of the original system. Additional larger tolerances can be obtained by the use of the relaxation strategy: as Section 5.4 shows, the allowable tolerances for $Ay = Bu^{(i)}_{k+j+1}$ are inversely proportional to the current eigenvalue residual norm. Therefore as the IRA method proceeds and converges to the desired invariant subspace, the relaxed tolerances keep increasing. Figure 5.4 shows the inner iteration counts required by four strategies with relaxed tolerances for solving $Ay = Bu^{(i)}_{k+j+1}$:

- "No Tuning-Fix" (dotted line) – Solve the original systems $Ay = Bu^{(i)}_j$ with preconditioned GMRES to the *fixed* tolerance $\delta_f = \frac{\epsilon}{2k}$. This performance of this strategy is already given in Figure 5.2; it is shown again to illustrate the performance improvement obtained by the following advanced strategies.

- "Orig_Tuning-Orig_Relax" ($\triangledown$, solid line) – Solve $Ay = Bu_{k+j+1}^{(i)}$ by GMRES with the original tuning to the *relaxed* tolerances δ_r given by the original estimate.
- "New_Tuning(5)-New_Relax" ($\diamondsuit$, dashed line) – Solve $Ay = Bu_{k+j+1}^{(i)}$ by the two-phase strategy to the new estimated tolerances δ_r in (5.35); tuning is constructed using solution vectors from the current and 5 previous IRA cycles.
- "New_Tuning(5)-New_Relax-Recycling" ($\bigstar$, dashed line) – Solve $Ay = Bu_{k+j+1}^{(i)}$ by by the two-phase strategy to the new estimated tolerances δ_r; in addition, subspace recycling is used to solve the correction equations.

It is clear from Figure 5.4 that the relaxed tolerances help gradually reduce the inner iteration counts to very small numbers (curves with $\triangledown$, $\diamondsuit$, and $\bigstar$).

Figure 5.4 also shows the effectiveness of subspace recycling. For Problems 1 and 2, the use of this technique reduces the inner iteration counts by 40%–50% in initial Arnoldi steps (compare curves with $\diamondsuit$ to those with $\bigstar$). For Problems 3 and 4, where the original tuning does not perform well (see Figure 5.2), subspace recycling still decreases the inner iteration counts by numbers commensurate to the dimensions of recycled subspaces. As we discussed, subspace recycling achieves this improvement because it helps reduce the initial latencies of inner iterations. In addition, 5.3.3 shows that the recycled subspaces obtained from one correction equation can be applied directly to the solution of the next equation, because the preconditioned system matrix is identical for the correction equations in all Arnoldi steps. This makes subspace recycling very cheap to use.

Table 5.2 summarizes the total inner iteration counts needed for each strategy for solving $Ay = B_{k+j+1}^{(i)}$ arising in inexact IRA. Here, "New Tuning + New Relaxation + Subspace Recycling" and "Original Tuning + Original Relaxation" are the most

	No Tuning	New Tuning	New Tuning New Relaxation	**New Tuning New Relaxation Subspace Recycling**	New Tuning Original Relaxation Subspace Recycling	Original Tuning	**Original Tuning Original Relaxation**
Prob 1	2687	2430	1560	787	806	1842	1184
Prob 2	5524	5090	3631	1401	1469	4549	3494
Prob 3(a)	10114	7780	5334	3930	4163	10263	7619
Prob 3(b)	9966	8129	5775	4641	4786	9889	7193
Prob 4(a)	17321	14294	9934	7584	7897	17299	11365
Prob 4(b)	21186	17099	12570	9446	9635	21072	14424

Table 5.2: Inner iteration counts for different solution strategy for each problem

efficient strategies in this chapter and [28] respectively. Clearly, the best approach is to combine the two-phase algorithm (with the new tuning), relaxation strategy and subspace recycling.

For all problems, we found that solution strategies with the new estimated allowable tolerances (5.35) require slightly smaller number of inner iterations as those needed for strategies with the original estimated tolerances. Table 5.2 shows that the new estimated tolerances help decrease the inner iteration counts by about 2%–5% (compare the "New Tuning + New Relaxation + Subspace Recycling" with "New Tuning + Original Relaxation + Subspace Recycling") when used with the two-phase strategy and subspace recycling. In fact, the new estimated tolerances tend to be a small multiple (say, 2 to 10) of the original estimated ones in most IRA cycles for all test problems.

Some heuristic remarks can be made for the two estimations. First, the substitution of $\min|\lambda(T_k) - \lambda(T_m^{22})|$ for $\text{sep}(T_m^{22}, T_k)$ in the original estimation seems reasonable, in the sense that the former is usually not obviously larger than the latter. In fact, in the setting of eigenvalue computation, we expect two basic properties to hold: (1) the desired Ritz vectors generated by the Rayleigh-Ritz procedure is not far from the best approximation available in subspace from which the Ritz vectors are extracted, and (2) small eigenvalue residual of the desired approximate invariant subspace implies good eigenvector approximation. However, by analogy to the results in [88] and Chapter 2 of [45], both properties may not be true if $\mathrm{sep}(T_m^{22}, T_k)$ is considerably smaller than $\|T_m^{12}\|$ or $\min|\lambda(T_k) - \lambda(T_m^{22})|$ in our context. In the usual situations when the two properties hold, $\min|\lambda(T_k) - \lambda(T_m^{22})|$ is expected to be not much larger than $\mathrm{sep}(T_m^{22}, T_k)$. Second, numerical evidences help to understand why the new estimate tends to be slightly larger than the original estimate. In fact, note that $\min|\lambda(T_k) - \lambda(T_m^{22})|$ and $\max|\lambda(T_k) - \lambda(T_m^{22})|$ are the smallest and largest eigenvalue of the Sylvester operator $\mathcal{S}_m$ $(G \to \mathcal{S}_m(G) : T_m^{22}G - GT_k)$; see [86], page 17. For the test problems with spectral transformation, it was consistently found that the largest eigenvalue of $\mathcal{S}_m$ is only about $10 - 100$ times of the smallest eigenvalue of $\mathcal{S}_m$, as long as the shift is not too close to an eigenvalue of the matrix pair (A, B). As the quantity in (5.33) used in the new estimation is always between the two extreme eigenvalues of $\mathcal{S}_m$ in practice, it is not unexpected that this quantity tends to be small multiple of $\min|\lambda(T_k) - \lambda(T_m^{22})|$. In conclusion, the original estimated allowable tolerance seems reasonably accurate for the test problems.

5.6 Concluding remarks

We have studied the inexact implicitly restarted Arnoldi (IRA) method for solving generalized eigenvalue problems with shift-invert and Cayley transformations, with focus on a few strategies that help reduce the inner iteration counts. We present a new tuning strategy using the solution vectors from the current and previous IRA cycles, and discuss a two-phase algorithm involving a correction equation for which the tolerance can be considerably bigger than that for the original system. In addition, subspace recycling can be used easily for the correction equation to further reduce the inner iteration counts. We analyze the allowable errors in Arnoldi steps and propose an alternative estimate of relaxed tolerances for the original linear systems. Numerical experiments show that the combined use of these strategies lead to significant speedup of inner iterations.

6 Conclusions and future work

This thesis is concerned with numerical solution of eigenvalue problems with spectral transformations. Specifically, we studied inexact Rayleigh quotient iteration (RQI), subspace iteration and implicitly restarted Arnoldi method (IRA) for computing one or a few eigenpairs of a matrix A or a matrix pencil (A, B), for which the shift-invert or Cayley transformations are used to map the desired eigenvalues to dominant ones of a transformed problem. Matrix-vector products involving the spectral transformations require solutions to corresponding shifted linear systems, which are done by iterative solvers when direct solvers are not an option, such as when the matrices are very large and sparse. In these cases, inexact eigenvalue algorithms with structure of "inner-outer" iteration need be used. Our major focus is on the study of a variety of techniques that help reduce the inner iteration counts without obviously affecting the convergence of the outer iteration for computing eigenvalues. The main results achieved in each chapter are summarized as follows.

- In Chapter 3, we present a detailed analysis of several versions of the MINRES algorithm for approximately solving the linear systems that arise when RQI is used to compute the lowest eigenpair of a symmetric positive definite matrix. We show that the initial slow convergence of MINRES does not affect the rate at which the MINRES iterate converge to the desired eigenvector, and this rate only depends on an effective condition number of the shifted coefficient matrix. We obtain a better understanding of the limitation of ordinary preconditioned MINRES in this context and the virtue of a new type of preconditioner with "tuning." A new tuning strategy based on a rank-2 modification of a preconditioner can be applied to guarantee positive definiteness of the tuned preconditioner.

- In Chapter 4, we study an inexact subspace iteration for solving generalized non-

Hermitian eigenvalue problems with spectral transformation. We provide new insights into the tuning strategy that has been studied for this algorithm applied to standard eigenvalue problems and propose an alternative way to use the tuned preconditioner to achieve similar performance for generalized problems. In addition, we show that the cost of inner iterations can be further reduced by using deflation of converged Schur vectors, special starting vectors constructed from previously solved linear systems, and iterative linear solvers with subspace recycling.

- In Chapter 5, we investigate an inexact IRA method for computing a few eigenpairs of generalized non-Hermitian eigenvalue problems with spectral transformation. We study a new tuning strategy constructed from solution vectors in both previous and the current IRA cycles, and show that tuning can be used in a new two-phase algorithm to greatly reduce inner iteration counts. We give a refined analysis of allowable errors for the inexact solves used in the Arnoldi steps, from which a new heuristic estimate of the allowable tolerances of inner solvers is proposed. In addition to the use of tuning and relaxed tolerances, the inner iteration cost can be further reduced by using subspace recycling with iterative linear solvers.

It should be noted that for both inexact Rayleigh quotient iteration and inexact subspace iteration, the inner solves done with tuning have close connections with solves performed by the basic Jacobi-Davidson method. In addition, we mentioned in Chapter 2 that the shift-invert residual Arnoldi method, a variant of the Arnoldi method based on Krylov subspaces, also has a few similarities to the Jacobi-Davidson method. Roughly speaking, the generic idea of these approaches is to solve a correction equation where the right-hand side is some variant of the eigenvalue residual vector of the desired eigenpairs, and then to add the computed correction to the current approximate eigenvector or to a subspace for the candidate approximate eigenvectors.

Other research directions related to inexact eigenvalue algorithms or other branches of scientific computing include:

- better refined allowable errors in Arnoldi steps of IRA; the error can be different for individual Arnoldi steps in a given IRA cycle;
- a complete theoretical and numerical comparison of the shift-invert residual Arnoldi method with the inexact IRA developed in this thesis;
- efficient strategies of subspace restarting and some techniques for further reducing the inner iteration counts for the shift-invert residual Arnoldi method;
- the relation between the errors of inner solves and the convergence of outer iterations for the full Jacobi-Davidson method;
- potential hybrid inexact methods that properly combine inexact eigenvalue algorithms that work with an increasing sequence of tolerances (e.g., subspace iteration and shift-invert residual Arnoldi method) and those working with a decreasing sequence of tolerances (e.g., IRA), so that the tolerance of inner solves may be kept modest throughout the progress of outer iterations;
- inexact eigenvalue algorithms for nonlinear problems, e.g., quadratic eigenvalue problems, and those with orthogonality constraints (see [3]);
- possible applications of preconditioners with low-rank modification and other ideas studied in this thesis to linear systems arising in different settings, for example, interior-point method for numerical optimization.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen., *LAPACK Users' Guide*, SIAM, Philadelphia, Third edition, 1999.

[2] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics, Vol. 9 (1951), pp. 17–29.

[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[4] C. A. Beattie, M. Embree, and D. C. Sorensen, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Review, Vol. 47, No. 3 (2005), pp. 492–515.

[5] J. Berns-Müller and A. Spence, *Inexact inverse iteration and GMRES*, Tech Report maths0507, University of Bath, Bath, UK, 2005

[6] J. Berns-Müller and A. Spence, *Inexact inverse iteration with variable shift for nonsymmetric generalized eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, Vol. 28, No. 4, (2006), pp. 1069–1082.

[7] J. Berns-Müller, I. G. Graham, and A. Spence, *Inexact inverse iteration for symmetric matrices*, Linear Algebra and its Applications, Vol. 416, No. 2–3, (2006), pp. 389-413.

[8] A. Bouras and V. Frayssé, *A relaxation strategy for the Arnoldi method in eigenproblems.* Technical Report TR/PA/00/16, CERFACS, Toulouse, France, 2000.

[9] A. Bouras and V. Frayssé, *A relaxation strategy for inner-outer linear solvers in domain decomposition methods.* Technical Report TR/PA/00/16, CERFACS, Toulouse, France, 2000.

[10] A. Bouras and V. Frayssé, *Inexact matrix-vector products in Krylov methods for solving linear systems: a relaxation strategy*, SIAM Journal on Matrix Analysis and Applications, Vol. 26, Issue 3 (2005), pp. 660–678.

[11] K. A. Cliffe, T. J. Garratt, and A. Spence, *Eigenvalues of block matrices arising from problems in fluid mechanics*, SIAM Journal on Matrix Analysis, Vol. 15, No. 4, (1994), pp. 1310–1318.

[12] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices*, Journal of Computational Physics, Vol. 17 (1975), pp 87–94.

[13] E. de Sturler, Private communication, October 2008.

[14] E. de Sturler, *Truncation strategies for optimal Krylov subspace methods*, SIAM Journal on Numerical Analysis, Vol. 36, No. 3, (1999), pp. 864–889.

[15] James W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[16] F. A. Dul, *MINRES and MINERR are better than SYMMLQ in eigenpair computations*, SIAM Journal on Scientific Computing, Vol. 19, No. 6, (1998), pp. 1767–1782.

[17] H. C. Elman, A. R. Ramage, D. J. Silvester, and A. J. Wathen, *Incompressible Flow Iterative Solution Software Package*, http://www.cs.umd.edu/ elman/ifiss.html.

[18] H. C. Elman, A. R. Ramage, D. J. Silvester, *IFISS: A Matlab toolbox for modelling incompressible flow*, ACM Transactions on Mathematical Software, Vol. 33, No. 2, (2007), Article 14, 18 pages.

[19] H. C. Elman, D. J. Silvester and A. J. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, New York, 2005.

[20] H. C. Elman, D. J. Silvester and A. J. Wathen, *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations*, Numerische Mathematik, Vol. 90, No. 4, (2002), pp. 665–688.

[21] O. G. Ernst, *Residual-minimizing Krylov subspace methods for stabilized discretizations of convection-diffusion equations*, SIAM Journal on Matrix Analysis and Applications, Vol. 21, No. 4 (2000), pp. 1079–1101.

[22] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson Style QR and QZ Algorithms for the Reduction of Matrix Pencils*, SIAM Journal on Scientific Computing, Vol. 20, No. 1 (1998), pp 94–125.

[23] R. Fletcher, *Conjugate gradient methods for indefinite systems*, Proceedings of the Dundee Biennial Conference on Numerical Analysis, 1974, G. A. Watson, ed., Springer-Verlag, New York, 1975, pp. 73–89.

[24] M. A. Freitag and A. Spence, *Convergence theory for inexact inverse iteration applied to the generalised nonsymmetric eigenproblem*, Electronic Transactions on Numerical Analysis, Vol. 28 (2007), pp. 40–64.

[25] M. A. Freitag and A. Spence, *Convergence rates for inexact inverse iteration with application to preconditioned iterative solves*, BIT Numerical Mathematics, Vol. 47, No.1 (2007), pp. 27–44.

[26] M. A. Freitag and A. Spence, *A tuned preconditioner for inexact inverse iteration applied to Hermitian eigenvalue problems*, IMA Journal on Numerical Analysis, Vol. 28, No. 3 (2007), pp. 522-551.

[27] M. A. Freitag and A. Spence, *Rayleigh quotient iteration and simplified Jacobi-Davidson method with preconditioned iterative solves*, Linear Algebra and its Applications, Vol. 428, No. 8–9 (2008), pp. 2049-2060.

[28] M. A. Freitag and A. Spence, *Shift-invert Arnoldi's method with preconditioned iterative solves*, to appear in SIAM Journal on Matrix Analysis and Applications.

[29] M. A. Freitag, A. Spence and E. Vainikko, *Rayleigh quotient iteration and simplified Jacobi-Davidson with preconditioned iterative solves for generalised eigenvalue problems*, submitted.

[30] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM Journal on Scientific Computing, Vol. 14, No. 2 (1993), pp. 470–482.

[31] R. W. Freund and N. M. Nachtigal *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numerische Mathematik, Vol. 60, No. 1, (1991), pp. 315–339.

[32] G. H. Golub and C. F. van Loan, *Matrix Computations, 3rd Edition*, The Johns Hopskins University Press, Baltimore, 1996.

[33] G. Golub and Q. Ye, *Inexact inverse iteration for generalized eigenvalue problems*, BIT Numerical Mathematics, Vol. 40, No. 4 (2000), pp. 671–684.

[34] G. Golub and Q. Ye, *An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, SIAM Journal on Scientific Computing, Vol. 24, No. 1, (2002), pp. 312–334.

[35] G. H. Golub, Z. Zhang, and H. Zha, *Large sparse symmetric eigenvalue problems with homogeneous linear constraints: the Lanczos process with inner-outer iterations*, Linear Algebra and its Applications, Vol. 309, No. 1–3 (2000), pp. 289–306.

[36] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[37] M. H. Gutknecht, *A completed theory of the unsymmetric Lanczos process and related algorithms: Part I*, SIAM Journal on Matrix Analysis and Applications, Vol. 13, No. 2 (1992), pp. 594–639.

[38] M. H. Gutknecht, *A completed theory of the unsymmetric Lanczos process and related algorithms: Part II*, SIAM Journal on Matrix Analysis and Applications, Vol. 15, No. 1 (1994), pp. 15–58.

[39] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, Vol. 49, No. 6 (1952), pp. 409–435.

[40] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, Vol. 34, No. 5 (1997), pp. 1911–1925.

[41] *HSL*, http://www.cse.scitech.ac.uk/nag/hsl/

[42] S. Kaniel, *Estimates for some computational techniques in linear algebra*, Mathematics of Computation, Vol. 20, No. 95 (1966), pp. 369–378.

[43] Y. Lai, K. Lin, and W. Lin, *An inexact inverse iteration for large sparse eigenvalue problems*, Numerical Linear Algebra with Applications, Vol. 4, No. 5 (1997), pp. 425–437.

[44] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau of Standards, Vol. 45, No. 4 (1950), pp. 255–282.

[45] C.R. Lee, *Residual Arnoldi Methods: Theory, Package and Experiments*, Ph.D thesis, Department of Computer Science, University of Maryland, 2007.

[46] C.R. Lee and G. W. Stewart, *Analysis of the Residual Arnoldi Method* , Technical Report, UMIACS TR-2007-45, CMSC TR-4890, University of Maryland, 2007.

[47] R. B. Lehoucq, *Implicitly restarted Arnoldi methods and subspace iteration*, SIAM Journal on Matrix Analysis and Applications, Vol. 23, No. 2 (2001), pp. 551–562.

[48] R. B. Lehoucq and J. A. Scott, *An evaluation of subspace iteration software for sparse nonsymmetric eigenvalue problems*, RAL-TR-96-022, Rutherford Appleton Lab., 1996.

[49] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.

[50] The MathWorks Inc., *MATLAB Online Documentation, R2009a*, The Mathworks Inc., Natick, MA, 2009.

[51] *Matrix Market*, http://math.nist.gov/MatrixMarket/

[52] K. Meerbergen and A. Spence, *Implicitly restarted Arnoldi with purification for the shift-invert transformation*, Mathematics of Computation, Vol. 66, No. 218 (1997), pp. 667–689.

[53] K. Meerbergen, A. Spence and D. Roose, *Shift-invert and Cayley transforms for detection of rightmost eigenvalues of nonsymmetric matrices*, BIT Numerical Mathematics, Vol. 34, No. 3 (1994), pp. 409–423.

[54] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra and its Applications, Vol. 154–156 (1991), pp. 289–309.

[55] R. B. Morgan, *GMRES with deflated restarting*, SIAM Journal on Scientific Computing, Vol. 24, No. 1 (2002), pp. 20–37.

[56] R. B. Morgan and D. S. Scott, *Preconditioning the Lanczos Algorithm for Sparse Symmetric Eigenvalue Problems*, SIAM Journal on Scientific Computing, Vol. 14, No. 3 (1993), pp. 585–593.

[57] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, 1996.

[58] Y. Notay, *Convergence analysis of inexact Rayleigh Quotient Iteration*, SIAM Journal on Matrix Analysis and Applications, Vol. 24, No.3 (2003), pp. 627–644.

[59] B. Nour-Omid, B. N. Parlett, T. Ericsson, and P. S. Jensen, *How to implement the spectral transformation.* Mathematics of Computation, Vol. 48, No. 178 (1987), pp. 663–673.

[60] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D thesis, University of London, 1971.

[61] M. L. Parks, E. De Sturler, G. Mackey, D. D. Johnson and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, Vol. 28, No. 5 (2006), pp. 1651–1674.

[62] Beresford N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.

[63] C. Paige, B. Parlett and H. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical Linear Algebra with Applications, Vol. 2, No. 2 (1995), pp. 115–134.

[64] C. Paige and M. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, Vol. 12, No. 4 (1975), pp. 617–629.

[65] M. Robbé, M. Sadkane and A. Spence, *Inexact inverse subspace iteration with preconditioning applied to non-Hermitian eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, Vol. 31, No. 1 (2009), pp. 92–113.

[66] J. Rommes, *Arnoldi and Jacobi-Davidson methods for generalized eigenvalue problems $Ax = \lambda Bx$ with singular B*, Mathematics of Computation, Vol. 77, No. 262 (2008), pp. 995–1015.

[67] U. Rüde and W. Schmid, *Inverse Multigrid Correction for Generalized Eigenvalue Computations*, Technical Report 338, Universität Augsburg, 1995.

[68] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd Edition, SIAM, Philadelphia, 2003.

[69] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, Div. of John Wiley & Sons, Inc., New York, 1992.

[70] Y. Saad, *On the rates of convergence of the Lanczos and the block Lansczos methods*, SIAM Journal on Numerical Analysis, Vol. 17, No. 5 (1980), pp. 687–706.

[71] Y. Saad, *Projection methods for solving large sparse eigenvalue problems*, in B. Kågström and A. Ruhe, editors, Matrix Pencils, page 121–144, Springer-Verlag, New York, 1983.

[72] Y. Saad, *Variations of Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra and Its Applications, Vol. 34 (1980), pp. 269–295.

[73] Y. Saad and M. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, Vol. 7, No. 3 (1986), pp. 856–869.

[74] V. Simoncini, *Variable accuracy of matrix-vector products in projection methods for eigencomputation*, SIAM Journal on Numerical Analysis, Vol. 43, No. 3 (2005), pp. 1155–1174.

[75] V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT Numerical Mathematics, Vol. 42, No.1 (2002), pp.159–182.

[76] V. Simoncini and D. B. Szyld, *On the occurrence of Superlinear Convergence of Exact and Inexact Krylov Subspace Methods*, SIAM Review Vol. 47, No. 2 (2005), pp. 247–272.

[77] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM Journal on Scientific Computing, Vol. 25, No. 2 (2003), pp. 454–477.

[78] V. Simoncini and D. B. Szyld, *Relaxed Krylov subspace approximation*, Proceedings in Applied Mathematics and Mechanics, Vol. 5, No. 1 (2005), pp. 797–800.

[79] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numerical Linear Algebra with Applications, Vol. 14, No. 1 (2007), pp. 1–59.

[80] G. L. G. Sleijpen and D. R. Fokkema, *BICGSTAB(l) for linear equations involving unsymmetric matrices with complex spectrum*, Electronic Transactions on Numerical Analysis, Vol. 1 (1993). pp. 11–32.

[81] G. L. G. Sleijpen and J. van den Eshof, *On the use of harmonic Ritz pairs in approximating internal eigenpairs*, Linear Algebra and its Applications, Vol. 358, No. 1–3 (2003), pp. 115–137.

[82] G. L. G. Sleijpen, J. van den Eshof and M. B. van Gijzen, *Relaxation strategies for nested Krylov methods*, Journal of Computational and Applied Mathematics, Vol. 177, No. 2 (2005), pp. 347–365.

[83] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, Vol. 17, No. 2 (1996), pp 401–425.

[84] P. Smit and M. H. C. Paardekooper, *The effects of inexact solvers in algorithms for symmetric eigenvalue problems*, Linear Algebra and its Applications, Vol. 287 (1999), pp. 337–357.

[85] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM Journal on Matrix Analysis and Applications, Vol. 13, No. 1 (1992), pp. 357–385.

[86] G. W. Stewart, *Matrix Algorithms Volumn II: Eigensystems*, SIAM, Philadelphia, 2001.

[87] G. W. Stewart, *A Residual Inverse Power Method*, Technical Report, UMIACS TR-2007-09, CMSC TR-4890, University of Maryland, 2007.

[88] G. W. Stewart, *A generalization of Saad's theorem on Rayleigh-Ritz approximations*, Linear Algebra and its Applications, Vol. 327, No. 1–3 (2001), pp. 115–119.

[89] G. W. Stewart, *An Unreliable Convergence Criterion for Arnoldi's Method*, Technical Report, CMSC TR-4938, University of Maryland, 2009.

[90] G. W. Stewart, *On the semidefinite B-Arnoldi Method*, Technical Report, CMSC TR-4939, University of Maryland, 2009.

[91] Lloyd N. Trefethen and Mark Embree, *Spectra and Pseudospectra*, Princeton University Press, Princeton, 2005.

[92] J. van den Eshof and G. L. G. Sleijpen, *Inexact Krylov subspace methods for linear systems*, SIAM Journal on Matrix Analysis and Applications, Vol. 26, No. 1 (2005), pp. 125–153.

[93] H. A. van der Vorst *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, Vol. 13, Issue 2 (1992), pp. 631–644.

[94] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.