Subspace Iteration on Steroids – A New Highly Parallel Non-Hermitian Eigensolver

Ping Tak Peter Tang, James Kestyn, Eric Polizzi

***Abstract*—Calculating portions of eigenvalues and eigenvectors of matrices or matrix pencils has many applications. An approach to this calculation for Hermitian problems based on a density matrix has been proposed in 2009 and a software package called FEAST has been developed. The density-matrix approach allows FEAST's implementation to exploit a key strength of modern computer architectures, namely, multiple levels of parallelism. Consequently, the software package has been well received and subsequently commercialized. A detailed theoretical analysis of Hermitian FEAST has also been established very recently. This paper generalizes the FEAST algorithm and theory, for the first time, to tackle non-Hermitian problems. Fundamentally, the new algorithm is basic subspace iteration or Bauer bi-iteration, except applied with a novel accelerator based on Cauchy integrals. The resulting algorithm retains the multi-level parallelism of Hermitian FEAST, making it a valuable new tool for large-scale computational science and engineering problems on leading-edge computing platforms.**

I. Introduction

General non-Hermitian eigenvalue problems arise in many important applications of applied sciences and engineering that include economic modeling, Markov chain modeling, structural engineering, fluid mechanics, material science, and more (see [1], [2] for example). Solving complex symmetric (still non-Hermitian) eigenvalue problems are crucial in modeling open systems based on the perfectly matched layer (PML) technique that is staple tool in electromagnetics [3], nanoelectronics [4], and micro electromechanical systems MEMS [5]. As a tool in numerical linear algebra, non-Hermitian eigensolvers are kernels to non-linear eigenvalue problems such as quadratic or polynomial eigenvalue problems [2], [6]. More generally, advances in high-performance and big-data computing will only increase the use for general eigenvalue solvers in areas such as bioinformatics, social network, data mining, just to name a few. Compared to the Hermitian case, the arsenal of solvers available for non-Hermitian eigenproblems are much more meager (see for example [7] and more discussions in the end). Any addition to the software toolbox for the general scientific computing is therefore always timely and welcome.

For eigenproblems of moderate size, robust solvers are well developed and widely available [8], [9] and are sometimes referred to as direct solvers [10]. These solvers typically calculate the entire spectrum of dense matrices or matrix pencils. In many large-scale applications, the underlying linear systems are typically large and sparse and that often only selected regions of the spectrum are of interest. A new approach for these calculations for Hermitian matrices and matrix pencils based on density matrices has been proposed recently [11]. From an algorithmic point of view, this new approach tries to compute an exact invariant subspace – which is the action of the density matrix – approximately, as opposed to, for example, Krylov subspace methods (see for example [12], [13], [14], [15]) or Jocobi-Davidson method [16], [17] which try to compute subspaces that approximate invariant subspaces in a certain sense. The density-matrix approach maintains a basis for a fixed-dimension subspace but updates it per iteration. In this view, it is similar to the non-expanding subspace version of an eigensolver based on trace minimization [18], [19] but with a different subspace update strategy. From an implementation point of view, this new approach is similar to spectral divide-and-conquer [20], [21] in that the calculation is expressed in terms of high-level building blocks that can much better exploit the advantages of modern computing architectures. In this case, the high-level building block is a numerical-quadrature based technique to approximate an exact spectral projector. This building block consists of solving independent linear systems, each for multiple right hand sides. A software package FEAST [22], for Hermitian eigenproblems, based on this approach has been made available since 2009 and has recently been adopted in the commercial package Intel® Math Kernel Library. A comprehensive theoretical analysis of Hermitian FEAST has been completed very recently [23] by two of the authors of this present work.

In this paper, we extend the FEAST algorithm and theory to tackle non-Hermitian eigenproblems. Similar to the Hermitian case, the non-Hermitian FEAST algorithm takes the form of standard subspace iteration in conjunction with the Rayleigh-Ritz procedure (see for example [10], page 157, or [2], page 115.) For non-Hermitian matrices, left and right eigenvectors are in general non-orthogonal and different. There are two natural generalizations of subspace iterations to handle this complication. A one-sided approach where one focuses on either the right or left invariant subspace, or a Bauer bi-iteration approach where both invariant subspaces are targeted simultaneously. The crucial ingredient is that the subspace iteration here is carried out on an approximate spectral projector obtained by numerical quadrature. Our analysis shows that the quadrature approximation perturbs the projector's eigenvalues but not the eigenvectors. Consequently, the convergence of subspace iteration can be established similar to the approaches shown in [2], suitably generalized as the left and right eigenspaces are now different. By exploring the structure of the generated subspaces, we show that the Rayleigh-Ritz procedure produces the targeted eigenpairs. Typical to many large-scale applications, the target eigenpairs are a small portion of the entire spectrum. In this case, the dominant work of our algorithm is the quadrature computation which possesses multiple levels of parallelism, making

Ping Tak Peter Tang, corresponding author Peter.Tang@intel.com is with Intel Corporation

James Kestyn and Eric Polizzi are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

this an excellent algorithm for high-performance computing. This paper aims to show how the various components of non-Hermitian FEAST fit together, with just enough mathematical analysis to make the presentation credible. A detailed numerical analysis similar to [23] for the Hermitian case is beyond the scope here. In subsequent sections we will describe the numerical-quadrature-based method to compute approximate spectral projector, establish the convergence of subspace iteration and bi-iteration with this approximate projector as an accelerator, prove convergence of the associated Rayleigh-Ritz procedure, and present numerical and performance examples.

II. Overview

To focus our discussion, we will concentrate on the simple eigenvalue problem (as opposed to the generalized eigenvalue problem of a matrix pencil) of a diagonalizable general matrix A. Section VIII will discuss FEAST's applicability beyond these assumptions. For the rest of this paper, A will be a $n \times n$ general matrix with the following eigendecomposition

$$A = X\Lambda Y^H,\ XY^H = I,$$

where Λ is a diagonal matrix of eigenvalues and X and Y are A's right and left eigenvectors, respectively:

$$X = [x_1, x_2, \ldots, x_n],\ Y = [y_1, y_2, \ldots, y_n],$$

$$Ax_j = \lambda_j x_j, \quad \text{and} \quad y_j^H A = \lambda_j y_j^H,$$

where Y^H stands for complex-conjugate transposition of Y. Unlike the Hermitian case where $A = A^H$, X and Y are not necessarily the same and do not have a standard normalization convention. It is customary to call the relationship $Y^H X = I$ as X and Y being bi-orthogonal.

Consider that the eigenvalues of interest, totaling m of them, are those that reside inside a simply connected domain $\mathcal{C}$ (e.g. disk, ellipse, etc.). Let $X_\mathcal{C}$ and $Y_\mathcal{C}$ be the corresponding sets of right and left eigenvectors. In particular, $X_\mathcal{C}$ and $Y_\mathcal{C}$ are $n \times m$ matrices with $Y_\mathcal{C}^H X_\mathcal{C} = I_m$. Our strategy is based on the use of the spectral projectors $X_\mathcal{C} Y_\mathcal{C}^H$ and $Y_\mathcal{C} X_\mathcal{C}^H$ which project to the right and left invariant subspaces, respectively. If we can compute $(X_\mathcal{C} Y_\mathcal{C}^H)u$ for any n-vector u, then applying $X_\mathcal{C} Y_\mathcal{C}^H$ on a set of random vectors[1] $U = [u_1, u_2, \ldots, u_p]$, $p \geq m$, will lead to $\mathrm{span}(X_\mathcal{C} Y_\mathcal{C}^H U) = \mathrm{span}(X_\mathcal{C})$ provided that $\mathrm{rank}(X_\mathcal{C} Y_\mathcal{C}^H U) = m$. A basis for $X_\mathcal{C}$ can then be constructed which in turn can be used (more details later) to compute the desired eigenvalues and right eigenvectors. The same approach using $Y_\mathcal{C} X_\mathcal{C}^H$ would lead us to the corresponding left eigenvectors.

The operator $X_\mathcal{C} Y_\mathcal{C}^H$ (and equally $Y_\mathcal{C} X_\mathcal{C}^H$) can be represented as a Cauchy integral (full details in Section III). Replacing this integral with a numerical quadrature rule yields a different operator that nevertheless approximates $X_\mathcal{C} Y_\mathcal{C}^H$. It turns out that applying this approximate spectral projector to vectors U is tantamount to solving a number of independent linear systems with U as right-hand sides; a procedure that is inherently parallel on a number of levels. Furthermore, the approximate spectral projector in fact preserves–exactly–the invariant subspaces $\mathrm{span}(X_\mathcal{C})$ and $\mathrm{span}(Y_\mathcal{C})$. Consequently, performing subspace iteration or Bauer bi-iteration with the approximate spectral projector becomes numerically effective as well as computationally efficient in capturing invariant subspaces as well as the associated eigenpairs. The general flow of the remaining sections is as follows. In Section III, we review the integral representations of $X_\mathcal{C} Y_\mathcal{C}^H$ and $Y_\mathcal{C} X_\mathcal{C}^H$ and analyze the properties of the quadrature-based operator. Based on the properties of the quadrature-based approximate spectral projectors, Section IV presents several variants of subspace iteration algorithms adapted for general non-Hermitian eigenvalue problems. We establish basic convergence properties of the subspaces generated by these iterative procedures. The structure of the subspaces generated in the iterative process are further analyzed in Section V, demonstrating their use in capturing the desired eigenpairs, which is the actual problem at hand. We present in Section VI a number of numerical experiments to illustrate the theoretical analysis. Scalability results are also presented, supporting our claim that this building block is a great addition to the overall toolbox for HPC calculation of non-Hermitian eigenvalue problems. In the concluding section, we put our new method in the context of other popular existing methods and share our views of future work.

III. Projection via Quadrature

In this section, we focus on the (right) spectral projector $X_\mathcal{C} Y_\mathcal{C}^H$. This can be viewed as a function of the matrix A, represented as a Cauchy integral.[2] We will show that when this integral is approximated by a numerical quadrature, the resulting approximate spectral projector has a number of useful properties. All discussions are applicable to the left projector, $Y_\mathcal{C} X_\mathcal{C}^H$, in a straightforward manner.

Let A be a diagonalizable matrix with spectral decomposition $A = X\Lambda Y^H$, $XY^H = I$. That is, X and Y are right and left eigenvectors and Λ is a diagonal matrix whose diagonal entries are the corresponding eigenvalues. Let the eigenvalues of interest reside in the interior of a simply connected region $\mathcal{C}$. We further assume that none of A's eigenvalues are on $\partial\mathcal{C}$, the boundary of $\mathcal{C}$. Let $\pi(\lambda)$ be the complex-valued function defined by the Cauchy integral (in the counter clockwise direction)

$$\pi(\lambda) = \frac{1}{2\pi\imath} \oint_{\partial\mathcal{C}} \frac{1}{z - \lambda}\, dz, \qquad \lambda \notin \partial\mathcal{C}. \tag{1}$$

The Cauchy integral theorem shows that $\pi(\lambda) = 1$ for λ inside the $\mathcal{C}$ and $\pi(\lambda) = 0$ for λ outside of $\mathcal{C}$. The function $\pi(\lambda)$ of a complex variable can be extended to a function of a matrix via

$$\pi(A) = \frac{1}{2\pi\imath} \oint_{\partial\mathcal{C}} (zI - A)^{-1}\, dz. \tag{2}$$

On the other hand, because $A = X\Lambda Y^H$, $XY^H = I$, one can see that

$$\pi(A) = X \left(\frac{1}{2\pi\imath} \oint_{\partial\mathcal{C}} (zI - \Lambda)^{-1}\, dz \right) Y^H = X_\mathcal{C} Y_\mathcal{C}^H.$$

[1] The effectiveness of randomized methods have been studied rigorously in a large body of recent works. Excellent surveys can be found in [24], [25].

[2] Representing functions of matrices as Cauchy integrals is well studied. The interested reader can find further details in [26], [27].

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

This is because the middle integral is simply a diagonal matrix whose diagonal entries are all zeros except for the 1's at the locations corresponding to $\lambda \in \mathcal{C}$. Equation 2 shows that the application of the projector $X_{\mathcal{C}}Y_{\mathcal{C}}^H$ to a set of vectors $U = [u_1, u_2, \ldots, u_p]$ admits an integral representation

$$(X_{\mathcal{C}}Y_{\mathcal{C}}^H)U = \frac{1}{2\pi\imath}\oint_{\partial\mathcal{C}} (zI - A)^{-1}U\, dz. \tag{3}$$

It is therefore natural to approximate the integral in Equation 3 by a quadrature rule. For the rest of the paper, we consider $\partial\mathcal{C}$ to be ellipses parametrized by

$$\phi(t) = c + R\left(\cos\left(\frac{\pi}{2}(1+t)\right) + \imath a \sin\left(\frac{\pi}{2}(1+t)\right)\right), \tag{4}$$

where $c \in \mathbb{C}$ is the center, with horizontal and vertical axes of lengths $R > 0$ and $Ra > 0$, respectively. We can apply any quadrature rule for integrating a function $f(t)$ on $[-1, 1]$ to obtain an approximation of $\pi(\lambda)$. Let $\int_{-1}^{1} f(t)dt \approx \sum_{k=1}^{q} w_k f(t_k)$ be a quadrature rule based on q pairs of $(\text{node}, \text{weight})$, $\{(t_k, w_k) | t_k \in [-1, 1], w_k > 0, k = 1, 2, \ldots, q\}$.

$$\begin{aligned}
\pi(\lambda) &= \frac{1}{2\pi\imath}\oint_{\partial\mathcal{C}} \frac{1}{z-\lambda}\, dz, \\
&= \frac{1}{2\pi\imath}\int_{-1}^{3} \frac{\phi'(t)}{\phi(t)-\lambda} dt, \\
&= \frac{1}{2\pi\imath}\int_{-1}^{1} \left(\frac{\phi'(t)}{\phi(t)-\lambda} + \frac{\phi'(2-t)}{\phi(2-t)-\lambda}\right).
\end{aligned}$$

Applying the quadrature rule[3] yields $\rho(\lambda) \approx \pi(\lambda)$,

$$\rho(\lambda) \stackrel{\text{def}}{=} \sum_{k=1}^{q} \frac{w_k}{2\pi\imath}\left(\frac{\phi'(t_k)}{\phi(t_k)-\lambda} + \frac{\phi'(\tilde{t}_k)}{\phi(\tilde{t}_k)-\lambda}\right), \quad \tilde{t}_k = 2 - t_k. \tag{5}$$

Applying Equation 5 to A results in an approximate spectral projection operation $\pi(A)U \approx \rho(A)U$, $\rho(A)U$ defined as

$$\sum_{k=1}^{q} \frac{w_k}{2\pi\imath}\left[\phi'(t_k)(\phi(t_k)I - A)^{-1} + \phi'(\tilde{t}_k)(\phi(\tilde{t}_k)I - A)^{-1}\right] U. \tag{6}$$

For a quadrature rule that uses neither -1 nor 1 as nodes, computing $\rho(A)U$ via Equation 6 involves solving $2q$ systems of linear equations, each with p right hand sides. For a quadrature rule with $t_1 = -1$ and $t_q = 1$, there are $2(q-1)$ linear systems to solve. Solutions of multiple independent linear systems for multiple right hand sides make $\rho(A)U$ a kernel operation with rich parallelism. Furthermore, $\rho(A)U$ is numerically effective, as we now explain.

Because $A = X\Lambda Y^H$, we have

$$\rho(A) = X\rho(\Lambda)Y^H. \tag{7}$$

In other words, while $\rho(A)$ is not exactly $X_{\mathcal{C}}Y_{\mathcal{C}}^H$, $\text{span}(X_{\mathcal{C}})$ and $\text{span}(Y_{\mathcal{C}})$ nevertheless remain to be right and left invariant subspaces of $\rho(A)$. Moreover, if $\rho(\lambda)$ maintains $|\rho(\lambda)| \approx 1$ for $\lambda \in \mathcal{C}$ and $|\rho(\lambda)| \ll 1$ for $\lambda \notin \mathcal{C}$, then $\rho(A)U$ would be effective in "tilting" U towards the direction of $\text{span}(X_{\mathcal{C}})$. We therefore examine the ratio of $|\rho(\mu)/\rho(\lambda)|$ for $\lambda \in \mathcal{C}$ and $\mu \notin \mathcal{C}$. To this end, if suffices to study the reference ρ function ρ_{ref} for the domain $\mathcal{C}$ that centers at the origin, with $R = 1$ because the $\rho(\lambda)$ function for an ellipse of a same "a" parameter but centered at c with "radius" R' is simply given $\rho(\lambda) = \rho_{\text{ref}}((\lambda - c)/R')$.

[3]See [28] for a different application of numerical quadrature to eigenvalue problems.

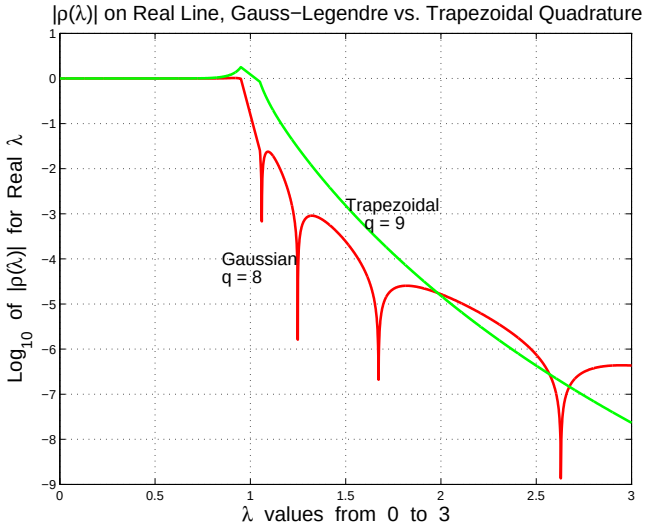

Fig. 1. For Hermitian eigenproblems, one only needs a $\rho_{\text{ref}}(\lambda)$ function that behaves well on the real line, namely, is close to 1 inside $[-1, 1]$ and small outside. Because of symmetry, this figure shows $|\rho_{\text{ref}}(\lambda)|$ only for $\lambda \geq 0$. For comparable computational cost of applying $\rho(A)$ to vectors U, Gauss-Legendre performs somewhat better than trapezoidal rule does.

To underline the difference between Hermitian and non-Hermitian problems, an effective quadrature rule for the former requires $|\rho_{\text{ref}}(\lambda)| \approx 1$ for $\lambda \in \mathcal{C}$ and $|\rho_{\text{ref}}(\lambda)| \ll 1$ for $\lambda \notin \mathcal{C}$ *only* for λ on the real line. Figure 1 shows $\log_{10}|\rho_{\text{ref}}(\lambda)|$ for real-valued λ for a Gauss-Legendre (with $q = 8$) and a trapezoidal rule (with $q = 9$). The percipitous drop of $|\rho_{\text{ref}}(\lambda)|$ for λ outside of $[-1, 1]$ signifies the effectiveness of quadrature-based approximate spectral projections.

For non-Hermitian problems, $\rho_{\text{ref}}(\lambda)$ has to "behave well" for λ in the complex plane. Consequently, for a given quadrature rule, we evaluate $\rho_{\text{ref}}(\lambda)$ at level curves similar to the boundary $\partial\mathcal{C}$:

$$\lambda(r, t) = r\left[\cos\left(\frac{\pi}{2}(1+t)\right) + \imath a \sin\left(\frac{\pi}{2}(1+t)\right)\right].$$

At each r below 1, $0 \leq r \leq 1 - \delta$ (δ set to 0.01), we record the minimum of $|\rho_{\text{ref}}|$ over the level curve, and at each $r \geq 1+\delta$, we record the maximum. That is, we examine the function

$$\eta(r) \stackrel{\text{def}}{=} \begin{cases} \min_t |\rho_{\text{ref}}(\lambda(r,t))| & \text{for } 0 \leq r \leq 1-\delta, \\ \max_t |\rho_{\text{ref}}(\lambda(r,t))| & \text{for } 1+\delta \leq r. \end{cases}$$

The function $\eta(r)$ serves as an indicator. An $\eta(r)$ that is close to 1 for $r < 1$ and very small for $r > 1$ corresponds to an approximate spectral projector that preserves the desired eigenspace well while attenuating the unwanted eigencomponents severely. Figure 2 shows three $\eta(r)$ functions, in logarithmic scale, corresponding to Gauss-Legendre quadrature ($q = 8$) on three different shapes of ellipses. Figure 3 shows different $\eta(r)$ functions, in logarithmic scale, corresponding to Gauss-Legendre and trapezoidal rules at different choices of q. The domain is set to be a circle. It is interesting to note that while Gauss-Legendre is in general a better choice for Hermitian problems (as Figure 1 suggests), trapezoidal rule

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

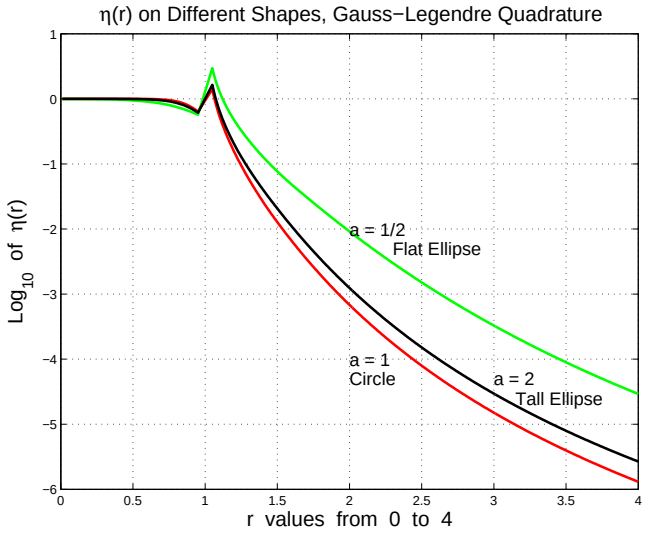

Fig. 2. These are the $\eta(r)$ functions corresponding to Gauss-Legendre quadrature with $q = 8$ nodes on $[-1, 1]$. We exhibit the result for three different elliptical domains. For simplicity, we employ circular domains for the rest of the paper, but different types of domains can be used. See further discussions in Section VIII.

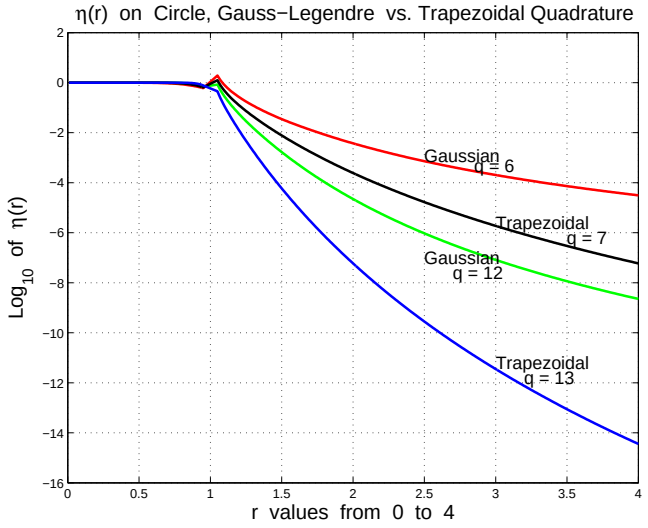

Fig. 3. This figure compares Gauss-Legendre quadrature to trapezoidal rule on a circular domain. Because trapezoidal rule uses both -1 and 1 as nodes on the integration interval $[-1, 1]$ while Gauss-Legendre uses neither, a q-node Gauss-Legendre and a $q+1$-node trapezoidal both require solving $2q$ linear systems when applying the spectral projector $\rho(A)$ to vectors U. The figure suggests that trapezoidal rule works better in general for non-Hermitian problems.

seems to fare better for non-Hermitian problems.[4]

IV. Subspace Iteration

Equation 7 in the previous section shows that the approximate spectral projector is of the form $\rho(A) = X\Gamma Y^H$ where $\Gamma = \rho(\Lambda)$. The invariant subspaces of $\rho(A)$ are identical to those of A. Moreover, A's eigenvalues inside $\mathcal{C}$ are mapped to the dominant values in Γ. Subspace iteration is a standard pedagogical method (see for example [10], [2]) that can be used to capture invariant subspaces. While subspace iteration is seldom used in practice in its pure form as the requirements for this method to succeed are stringent, the previous analysis on $\rho(A)$ suggests it to be a perfect candidate for this simple iterative method to converge rapidly. This is because the invariant subspace of interest corresponds to highly dominant eigenvalues. Algorithm SIR below is a straightforward adaptation of subspace iteration using $\rho(A)$ instead of A.

Algorithm SIR (Subspace Iteration for Right Eigenspace)

1: Pick random orthogonal $U_{(0)} \in \mathbb{C}^{n\times p}$, $U_{(0)}^H U_{(0)} = I$.
2: // In general, $p \ll n$.
3: Set $k \leftarrow 1$.
4: **repeat**
5: $\quad U_{(k)} \leftarrow$ orthonormalize$(\rho(A) \cdot U_{(k-1)})$
6: $\quad$ // Orthonormalization is a numerical safeguard.
7: $\quad k \leftarrow k + 1$
8: **until** Appropriate stopping criteria

While basic convergence properties about the sequence of subspaces generated by subspace iteration of a Hermitian matrices is easily available (see for example [2], [29]), generalized results for non-Hermitian case is more obscure.[5] Moreover, Algorithm SIR utilizes an approximate spectral projector, which leads to properties not shared by general subspace iterations.[6] In the remaining part of this section, we present several basic convergence properties of the generated subspace by Algorithm SIR as well as by other variants of SIR.

We start by numbering the γ_js (the eigenvalues of $\rho(A)$) so that

$$|\gamma_1| \geq |\gamma_2| \geq \cdots \geq |\gamma_n|,$$

and number A's eigenpairs accordingly:

$$\gamma_j = \rho(\lambda_j), \quad Ax_j = \lambda_j x_j, \quad y_j^H A = \lambda_j y_j^H, \quad j = 1, 2, \ldots, n.$$

The eigenpairs of interest are now in the beginning part of this numbering convention. Our analysis involves examining sections of the right and left eigenvectors X and Y, and the corresponding eigenvalues. We set up some simplifying notations: For integer ℓ, $1 \leq \ell \leq n$,

$$\begin{array}{ll} X_\ell = [x_1, x_2, \ldots, x_\ell], & X_{\ell'} = [x_{\ell+1}, x_{\ell+2}, \ldots, x_n], \\ Y_\ell = [y_1, y_2, \ldots, y_\ell], & Y_{\ell'} = [y_{\ell+1}, y_{\ell+2}, \ldots, y_n], \\ \Gamma_\ell = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_\ell), & \Gamma_{\ell'} = \mathrm{diag}(\gamma_{\ell+1}, \gamma_{\ell+2}, \ldots, \gamma_n). \end{array}$$

In particular, because X and Y are bi-orthogonal, we have $I = XY^H = X_\ell Y_\ell^H + X_{\ell'} Y_{\ell'}^H$, $Y_\ell^H X_\ell = I_\ell$, $Y_{\ell'}^H X_{\ell'} = I_{n-\ell}$, $Y_\ell^H X_{\ell'} = 0$, and $Y_{\ell'}^H X_\ell = 0$. We also normalize the lengths of the right and left eigenvectors so that $\|x_j\|_2 = \|y_j\|_2$ for $j = 1, 2, \ldots, n$. This normalization, while not crucial, is convenient for subsequent discussions.

Theorem 1. *Consider Algorithm SIR. Let $|\gamma_p| > |\gamma_{p+1}|$ and that the randomly chosen $U_{(0)}$ yields an invertible $Y_p^H U_{(0)}$. Then there are constants α_j, $j = 1, 2, \ldots, p$, such that for each $j = 1, 2, \ldots, p$, the generated subspace* $\mathrm{span}(U_{(k)})$ *at*

[4]Assuming no information of the eigenvalues' distribution is available a priori.

[5]This may be due to the fact that subspace iteration is usually not considered a powerful method for Hermitian problems, not to mention the more difficult non-Hermitian problems.

[6]Readers acquainted with acceleration methods can also view approximate spectral projection as a special acceleration, which warrant further analysis beyond the readily available results for plain subspace iteration alone.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

every iteration k contains a vector of the form $x_j + X_{p'} e_j^{(k)}$ where $e_j^{(k)} \in \mathbb{C}^{n-p}$ and satisfies $\|e_j^{(k)}\|_2 \leq \alpha_j |\gamma_{p+1}/\gamma_j|^k$.

There is an obvious left eigenspace variant Algorithm SIL of Algorithm SIR. One simply replace (for notational clarity) the letter U with V, and compute in Step 5

$$V_{(k)} \leftarrow \text{orthonormalize}((\rho(A))^H V_{(k-1)}).$$

Theorem 1 has an obvious version for SIL. Namely, there are constants β_j, $j = 1, 2, \ldots, p$, such that vectors of the form $y_j + Y_{p'} f_j^{(k)}$ reside in $\text{span}(V_{(k)})$ where

$$\|f_j^{(k)}\|_2 \leq \beta_j |\gamma_{p+1}/\gamma_j|^k$$

for each left eigenvector y_j, $j = 1, 2, \ldots, p$ and each iteration k. The proof is trivially similar to the one for Theorem 1.
SIR and SIL can be combined to target both the right and left eigenspaces simultaneously. This is tantamount to generalizing the classical Bauer bi-iteration [30], [31] to one that uses an approximate spectral projector. The resulting Algorithm BIT is outlined here.

Algorithm BIT (Bi-Iteration with Spectral Projector)

1: Pick random $U_{(0)}, V_{(0)} \in \mathbb{C}^{n \times p}$, with $V_{(0)}^H U_{(0)} = I$.
2: Set $k \leftarrow 1$.
3: **repeat**
4: $\quad (\tilde{U}, \tilde{V}) \leftarrow (\rho(A) \cdot U_{(k-1)},\ \rho(A)^H \cdot V_{(k-1)})$
5: $\quad (U_{(k)}, V_{(k)}) \leftarrow$ bi-orthogonalize$(\tilde{U}, \tilde{V})$.
6: $\quad$ // This makes $V_{(k)}^H U_{(k)} = I_p$.
7: $\quad k \leftarrow k + 1$
8: **until** Appropriate stopping criteria

Mathematically, the subspaces spanned are unchanged when their basis vectors are orthonormalized or bi-orthogonalized. Thus, the space $\text{span}(U_{(k)})$ that corresponds to Step 4 of BIT is the same as the one corresponds to Step 5 in SIR, had both SIR and BIT started with the same random subspace of $\text{span}(U_{(0)})$. The same holds for the left subspaces $\text{span}(V_{(k)})$. Consequently, the convergence properties described in Theorem 1 for the right subspaces and the analogue for the left subspaces hold *simultaneously* in Algorithm BIT. The generated subspaces, as represented by the basis vectors $U_{(k)}$ and $V_{(k)}$ in the various variants of subspace iteration algorithms SIR, SIL, and BIT can be used in a Rayleigh-Ritz manner to yield good approximations to the targeted eigenpairs. This is the subject of the next section.

V. Non-Hermitian FEAST

Intuitively, under favorable conditions, the orthogonal basis obtained in Algorithm SIR will have captured the right eigenvectors $x_1, x_2, \ldots, x_p$ quite accurately after just a small number of iterations. Consequently, solving the eigensystem of the (small) $p \times p$ system $\tilde{A} = U^H A U$ for eigenpairs $(\tilde{\Lambda}, \tilde{X})$ should yield $\tilde{\Lambda} \approx \Lambda_p$ and $U\tilde{X} \approx X_p$ (up to permutations). This is the basic form of the "one-sided" FEAST algorithm for non-Hermitian problem, which we call R-FEAST for right-sided FEAST.

In this section, we first outline the basic analysis on why Algorithm R-FEAST would work under very favorable conditions. Using this as a building block, we show that the algorithm would work under more realistic conditions.

Algorithm R-FEAST (Right-sided FEAST)

1: Pick random orthogonal $U_{(0)} \in \mathbb{C}^{n \times p}$, $U_{(0)}^H U_{(0)} = I$.
2: Set $k \leftarrow 1$.
3: **repeat**
4: $\quad \tilde{U} \leftarrow$ orthonormalize$(\rho(A) \cdot U_{(k-1)})$
5: $\quad \tilde{A} \leftarrow \tilde{U}^H A \tilde{U}$
6: $\quad$ Solve eigenproblem $\tilde{A}\tilde{X} = \tilde{X}\tilde{\Lambda}$ for $(\tilde{\Lambda}, \tilde{X})$
7: $\quad$ Set $U_{(k)} \leftarrow \tilde{U}\tilde{X}$ and $\Lambda_{(k)} \leftarrow \tilde{\Lambda}$.
8: $\quad k \leftarrow k + 1$
9: **until** Appropriate stopping criteria

Finally, similar analysis is carried out to Algorithm Bi-FEAST, which is the natural R-FEAST analogue based on two-sided iteration BIT.
We adopt in our analysis a simplified approach and express resulting mathematical upper bounds in the form of $O(\epsilon)$ for small $\epsilon > 0$ – despite the derivations themselves necessarily involve extra details such as $\|A\|_2$, condition number of the matrices X, X_m, $X_{m'}$, and their respective norms, etc. These details also underline the mathematical complications that correspond to non-Hermitian problems. Had the problem be Hermitian, that X and Y are the same and unitary renders virtually all related norms and condition numbers unity. The purpose of this paper is to give an overall picture of non-Hermitian FEAST and a high-level view of how and why it works, corroborated by numerical illustration. A detailed error analysis would require a dedicated and much longer paper.

Theorem 2. *Let $Q = X_p R_p$ be orthogonal where X_p are the first p right eigenvectors. Let $\tilde{X}_p = X_p + X_{p'} E$ be a small perturbation of X_p where $\|X_{p'} E\|_2 \leq \epsilon \|X_p\|_2$, and $\|E\|_2 \leq \epsilon$ for some small ϵ. Then there exist a $\tilde{R}_p$ close to R_p so that $\tilde{Q} = \tilde{X}_p \tilde{R}_p$ is orthogonal. In particular, $\tilde{R}_p = R_p(1 + \Delta_R)$, $\|\Delta_R\|_2 = O(\epsilon)$.*

Theorem 3. *Let $Q = X_p R_p$ and $\tilde{Q} = \tilde{X}_p \tilde{R}_p$ be as described in Theorem 2. Then $\|Q^H A Q - \tilde{Q}^H A \tilde{Q}\|_2 = O(\epsilon)$.*

Theorem 3 shows why R-FEAST can yield the p eigenvalues Λ_p, at least under favorable conditions. Suppose p is such that $|\gamma_{p+1}/\gamma_p|$ is reasonably small (for example, below .1), then a moderate number of iterations of R-FEAST, because of Theorem 1, would produce an orthogonal basis U that contains vectors of the form $\tilde{X}_p = X_p + X_{p'} E$ that fit the description of Theorem 2. The ϵ will be of the form $O(|\rho(\lambda_{p+1})/\rho(\lambda_p)|^k)$, k being the iteration number. The orthogonal basis $\tilde{Q} = \tilde{X}_p \tilde{R}_p$ described in Theorem 2 must therefore be equivalent to U, and thus the eigenvalues of $\tilde{Q}^H A \tilde{Q}$ are exactly the $\tilde{\Lambda}$ computed in Step 6 of R-FEAST. On the other hand, the p exact eigenvalues of A, Λ_p, are those of the $p \times p$ system $Q^H A Q$ because Q is an orthogonal basis of $\text{span}(X_p)$. Therefore, $\tilde{\Lambda}$ are the eigenvalues of $\tilde{Q}^H A \tilde{Q}$, which is a slight perturbation of $Q^H A Q$, whose eigenvalues are Λ_p. In other words, R-FEAST is computing the exact solutions of a slightly perturbed problem.
The requirement of $|\gamma_{p+1}/\gamma_p|$ being small (a gap between γ_{p+1} and γ_p) is restrictive, and in fact unnecessary. The property of $\rho(\lambda)$ guarantees that for a given m, as long as $p \geq m$ is large enough, $|\gamma_{p+1}/\gamma_j| \ll 1$ for $j = 1, 2, \ldots, m$. It is not necessary to have a gap between γ_{p+1} and γ_p where

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

p is the dimension of the subspaces maintained by the algorithms here. In essence, the p-dimensional subspaces, while not necessarily capturing all the eigenvectors x_1 through x_p, will nevertheless be capturing x_1 through x_m, which should be sufficient if they already include all of the ones we are interested in. Therefore among the p eigenvalues computed in Step 6 of R-FEAST, m of them should be close to λ_1 through λ_m. The following theorem affirms this general picture.

Theorem 4. *Let $\tilde{Q}$ be $p > m$ orthogonal vectors with the first m columns of the form $\tilde{X}_m\tilde{R}_m$, $\tilde{X}_m = X_m + X_{m'}E$, $X_{m'}E$ small as described in Theorem 2, $\|E\|_2 \le \epsilon$ and $\|X_{m'}E\|_2 \le \epsilon\|X_m\|_2$. Partition the $p \times p$ matrix $\tilde{Q}^H A\tilde{Q}$ as*

$$\tilde{Q}^H A\tilde{Q} = \left[\begin{array}{c|c} A'_{11} & A'_{12} \\ \hline A'_{21} & A'_{22} \end{array}\right], \tag{8}$$

A'_{11} is $m\times m$ and A'_{21} is $(p-m)\times m$. Then $\|A'_{21}\|_2 = O(\epsilon)$.

Theorem 4 shows that R-FEAST works in general. Suppose p is chosen large enough so that $|\gamma_{p+1}/\gamma_m| \ll 1$ for some $m \le p$ where the desired eigenvalues are among the first m eigenvalues λ_1 through λ_m. Provided the eigenvectors are not too ill-conditioned, then as iterations proceed, the reduced systems share the same eigenvalues to a matrix of the form of Equation 8. Because A'_{21} is small (Theorem 4) and A'_{11} is a small perturbation of Q^HAQ, Q being an orthogonal basis to $\mathrm{span}(X_m)$ (Theorem 3), m of the eigenvalues found in R-FEAST's Step 6 should approximate Λ_m, which in turn contains the targeted eigenvalues of interest.

Algorithm Bi-FEAST (Two-sided FEAST)

1: Pick random $U_{(0)}, V_{(0)} \in \mathbb{C}^{n\times p}$, with $V_{(0)}^H U_{(0)} = I$.
2: Set $k \leftarrow 1$.
3: **repeat**
4: $\quad(\tilde{U}, \tilde{V}) \leftarrow (\rho(A)\cdot U_{(k-1)},\ \rho(A)^H \cdot V_{(k-1)})$.
5: $\quad(\tilde{U}, \tilde{V}) \leftarrow$ bi-orthogonalize$(\tilde{U}, \tilde{V})$.
6: $\quad$// That is, $\tilde{V}^H\tilde{U} = I_p$.
7: $\quad\tilde{A} \leftarrow \tilde{V}^H A\tilde{U}$
8: $\quad$Solve eigenproblem $\tilde{A} = \tilde{X}\tilde{\Lambda}\tilde{Y}^H$ for $(\tilde{\Lambda}, \tilde{X}, \tilde{Y})$
9: $\quad$Set $U_{(k)} \leftarrow \tilde{U}\tilde{X}$, $V_{(k)} \leftarrow \tilde{V}\tilde{Y}$, and $\Lambda_{(k)} \leftarrow \tilde{\Lambda}$.
10: $\quad k \leftarrow k + 1$
11: **until** Appropriate stopping criteria

Just as Algorithm R-FEAST is Algorithm SIR enhanced with Rayleigh-Ritz, Algorithm Bi-FEAST on the next column is the result of adding Rayleigh-Ritz to Algorithm BIT. Laux [32] experimented with Bi-FEAST with some success, but expressed his need for theoretical backing. This need is now fulfilled by Theorems 2–4 for R-FEAST, as well as the next three theorems, for Bi-FEAST.

Theorem 5. *Let X_p and Y_p be the first p right and left eigenvectors. Let $\tilde{X}_p$ and $\tilde{Y}_p$ be small perturbations in the form $\tilde{X}_p = X_p + X_{p'}E$ and $\tilde{Y}_p = Y_p + Y_{p'}F$, $\|E\|_2, \|F\|_2 \le \epsilon$, $\|X_{p'}E\|_2 \le \epsilon\|X_p\|_2$, and $\|Y_{p'}F\|_2 \le \epsilon\|Y_p\|_2$. Then there exist $\tilde{R}_p = I_p + \Delta_R$, $\tilde{S}_p = I_p + \Delta_S$ close to I_p such that $\tilde{U} = \tilde{X}_p\tilde{R}_p$, $\tilde{V} = \tilde{Y}_p\tilde{S}_p$ are bi-orthogonal $\tilde{V}^H\tilde{U} = I_p$. In particular, $\|\Delta_R\|_2, \|\Delta_S\|_2 = O(\epsilon^2)$.*

Theorem 6. *Let $\tilde{U} = \tilde{X}_p\tilde{R}_p$ and $\tilde{V} = \tilde{Y}_p\tilde{S}_p$ be as described in Theorem 5. Then $\|\tilde{V}^H A\tilde{U} - \Lambda_p\|_2 = O(\epsilon^2)$.*

Theorems 5 and 6 show that if $|\rho(\lambda_{p+1}/\lambda_p|$ is reasonably small, one would expect the "reduced system" $\tilde{A}$ (Step 7 of Bi-FEAST) be similar[7] to a matrix of the form $\Lambda_p + O(\epsilon^2)$, which is a better approximation obtainable by one-sided iterations. Under the more realistic situation, one can only expect $|\rho(\lambda_{p+1})/\rho(\lambda_j)| \ll 1$ for $j = 1, 2, \ldots, m$ for some $m \le p$ (not necessarily all the way up to $m = p$). In this case, as in R-FEAST, we can expect m of $\tilde{A}$'s p eigenvalues to approximate Λ_p. This is formalized below by Theorem 7, which mirrors Theorem 4.

Theorem 7. *Let $p > m$ and that $\tilde{U}$, $\tilde{V}$ are bi-orthogonal with their first m columns of the form $\tilde{X}_m\tilde{R}_m$ and $\tilde{Y}_m\tilde{S}_m$, close to X_m and Y_m, as in Theorem 5. Partition the $p \times p$ matrix $\tilde{V}^H A\tilde{U}$ as*

$$\tilde{V}^H A\tilde{U} = \left[\begin{array}{c|c} A'_{11} & A'_{12} \\ \hline A'_{21} & A'_{22} \end{array}\right], \tag{9}$$

A'_{11} is $m \times m$. Then $\|A'_{21}\|_2, \|A'_{12}\|_2 = O(\epsilon)$.

Standard perturbation theory (see for example Theorem 2.8 in [33]) shows that, under mild assumptions, eigenvalues of the matrix in Equation 9 are those of $A'_{11} + O(\epsilon^2)$ and $A'_{22} + O(\epsilon^2)$. Theorems 5–7 lay the foundation for Bi-FEAST: In the absence of ill conditioning, one expects m of the p eigenvalues in $\Lambda_{(k)}$ (Step 9) will converge to Λ_m linearly at the rate $|\rho(\lambda_{p+1})/\rho(\lambda_m)|^{2k}$. While this is faster than the rate of $|\rho(\lambda_{p+1})/\rho(\lambda_m)|^k$ achievable by R-FEAST, the word of caution is that R-FEAST is inherently more stable. This is because Step 5 of R-FEAST utilizes a unitary transform, whereas the transform $\tilde{V}A\tilde{U}$ in Bi-FEAST's Step 7 can be much less stable (cf. Example 2 in Section VI).

VI. Numerical Experiments

The first three experiments in this section serve to illustrate the various convergence properties of non-Hermitian FEAST and were run with matlab. The last one is extracted from an actual application in electronic structure calculation and was run on a cluster (detials later). The first three examples use the matrices arise in quantum chemistry [34] called QC324 and QC2534 from the University of Florida collection [35]. These two matrices are similar in properties but differ in size. Figure 4 profiles the location of the eigenvalues in the complex plane. The last example uses two matrices of different sizes from a common application problem. This example illustrates the basic scalability properties of FEAST due mainly to the many parallel execution opportunities available in computing the quadrature-based approximate spectral projections.

For each of the first three tests, a circular domain is picked and one or several variants of FEAST are carried out with a specific subspace dimension p, which presumably is set to be moderately bigger then the number of eigenvalues expected inside the domain in question. Section VIII will discuss the choice of p further. During the iterations, we monitor the p eigenpairs computed from the reduced system (in Step 7 of R-FEAST, for example). A particular (right) eigenpair $(\tilde{\lambda}_j, u_j)$ is considered a candidate if $\tilde{\lambda}_j \in \mathcal{C}$ and the residual is reasonably small, typically, residual$_j \le 10^{-4}$ where

$$\text{residual}_j \stackrel{\text{def}}{=} \|Au_j - \tilde{\lambda}_j u_j\|_2/\|u_j\|_2.$$

[7]Two matrices B and C are similar if $B = ZCZ^{-1}$ for some Z.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

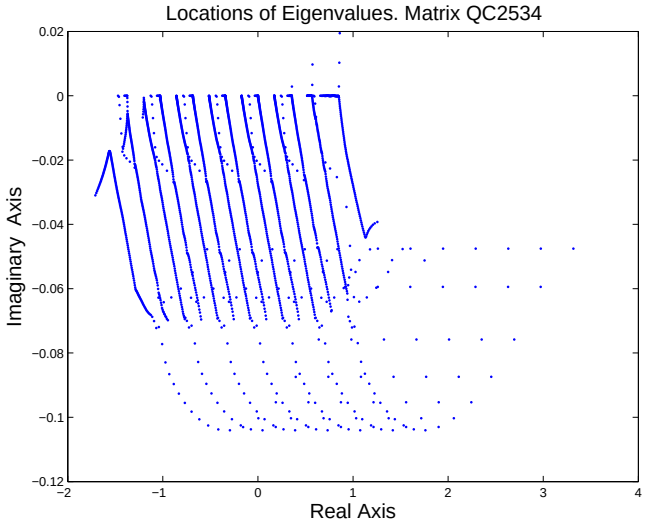

Fig. 4. Eigenvalues of the matrix QC2534 from University of Florida collection. Eigenvalues distribution of QC234 bears a resemblance.

To track convergence, we monitor the maximum of the residuals $\max_j(\text{residual}_j)$. We also monitor the change in trace $\stackrel{\text{def}}{=} \sum_j \tilde{\lambda}_j$ to gauge if the eigenvalues are converging to a certain level. While one should expect the maximum residual to settle to $O(\epsilon\|A\|_2)$ for machine epsilon ϵ, one cannot in general expect the change in trace to settle down to the same level: It is well known that, unlike Hermitian eigenvalues which are perfectly conditioned, a simple eigenvalue λ_j of a non-Hermitian A can change (see [36], [33]) by $O(\|x_j\|_2\|y_j\|_2\delta)$ when A is perturbed by $O(\delta)$.

Example 1:Simple Convergence of R-FEAST: We illustrate the most basic convergence properties with the small (dimension 324) matrix QC324. The domain $\mathcal{C}$ chosen is the disk of radius 0.01 centered on the real axis at -0.5, containing $m = 8$ eigenvalues. We employ Gauss-Legendre quadrature and picked $p = 8$. With this choice, $\log 10|\rho(\lambda_{p+1})/\rho(\lambda_m)| = -0.93$. The table here exhibits the expected behavior from both R-FEAST and Bi-FEAST. The eigenvalue and residual convergence rate are linear at roughly 0.9 digits per iteration, except that eigenvalues in Bi-FEAST converge as fast as 2×0.9 digits per iteration.

$p = m = 8$, Gauss-Legendre with $q = 8$

	$\log_{10}$|change in trace|		$\log_{10}$(max of residual)	
Iter.	R-FEAST	Bi-FEAST	R-FEAST	Bi-FEAST
4	-5.4	-0.0	-5.3	-4.8
5	-6.4	-8.6	-6.2	-5.7
6	-7.3	-10.5	-7.1	-6.6
7	-8.2	-12.4	-8.1	-7.6
8	-9.2	-14.3	-9.0	-8.5
9	-10.2	-14.4	-9.9	-9.4
10	-11.1	-14.5	-10.9	-10.3
11	-12.1	-15.1	-11.8	-11.3
12	-13.1	-14.8	-12.7	-12.2
13	-14.1	-14.8	-13.7	-13.1
14	-14.7	-14.5	-14.6	-14.1

Example 2: R-FEAST and Bi-FEAST: This example illustrates the sensitive nature of Bi-FEAST. We have seen in the previous example that Bi-FEAST can offer a faster convergence on the eigenvalues. But as discussed in Section V, Bi-FEAST is more sensitive to the conditioning of the eigenvalues. This is the case for the matrix QC2534 when the region is chosen to be the disk of radius 0.01 centered on the real axis at 0.85, containing 10 eigenvalues. With p set to $p = m + 5 = 15$, Gauss-Legendre quadrature with $q = 8$ yields $\log 10|\rho(\lambda_{p+1})/\rho(\lambda_m)| = -2.61$. The condition of the eigenvalues, however, are poor: the products $\|x_j\|\|y_j\|$ are of the order of 10^{11}. The table here shows that indeed the eigenvalues cannot be resolved to be much better than 5 or 6 digits. R-FEAST is able to deliver small residuals, while Bi-FEAST is hampered by the poor conditioning, as it is difficult to maintain bi-orthogonality between the x_j and y_j to full machine precision, precisely because their norms are large.

$p = m + 5 = 15$, Gauss-Legendre with $q = 6$

	$\log_{10}$|change in trace|		$\log_{10}$(max of residual)	
Iter.	R-FEAST	Bi-FEAST	R-FEAST	Bi-FEAST
2	0.6	1.0	-8.3	-4.1
3	-0.0	-5.3	-11.4	-5.8
4	-5.2	-5.3	-14.0	-5.9
5	-6.8	-5.4	-14.2	-6.2
6	-6.9	-5.3	-14.2	-5.9
7	-7.6	-5.3	-14.2	-6.1
8	-6.9	-5.6	-14.2	-6.0
9	-6.6	-5.4	-14.3	-6.0
10	-6.8	-5.7	-14.1	-5.8

Example 3: Different Quadratures: Figure 3 in Section III suggests that trapezoidal rule may work better in general. This example is consistent with this view, but illustrates some subtlety. Figure 3 depicts minimal convergence rate. Depending on the exact location of the eigenvalues, which is problem specific, a quadrature with a lower minimal convergence rate may actually still converge faster. Here we compute the eigenvalues of QC2534 that reside inside the disk of radius 0.02, centered on the real line at -0.17, which contains 28 eigenvalues. At each of two different settings, the table below exhibits the residual convergence for both Gauss-Legendre and trapezoidal quadrature. The behavior below is consistent with the actual values of $|\rho(\lambda_{p+1})/\rho(\lambda_m)|$.

R-FEAST, Gauss-Legendre(GL) vs. Trapezoidal(TR)

	$\log_{10}$(max of residual)			
	$p = m + 3 = 31$		$p = m + 6 = 34$	
Iter.	GL-8 nodes	TR-9 nodes	GL-8 nodes	TR-9 nodes
2	-4.1	-4.0	-4.6	-5.7
3	-5.6	-5.4	-6.4	-8.2
4	-7.0	-6.8	-8.3	-11.3
5	-8.7	-8.2	-10.1	-13.8
6	-10.9	-9.4	-11.9	-14.2
7	-12.9	-10.6	-13.7	-14.3
8	-14.1	-11.9	-14.4	-14.4

The typical convergence pattern of the residuals is as follows. The subspace dimension p is in general bigger than the number of eigenvalues inside the targeted domain. Some of the residuals that are not targeted (we usually call them collaterals) will converge slowly, or not at all. Figure 5 displays the residuals of our current QC2534 test using Gauss-Legendre with p set to $m + 6$. Notice that the 28 targeted residuals converge linearly at the expected rate. Convergence of the collaterals are much slower, and some not at all.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

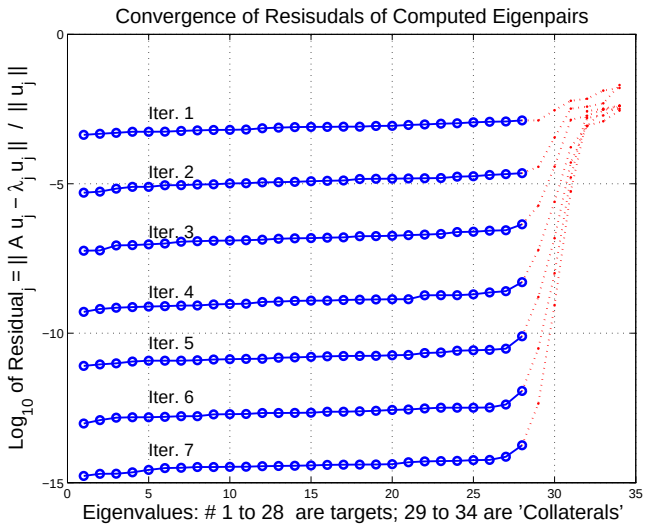

Fig. 5. Convergence of residuals: targeted and the "collaterals". Residuals are sorted to give a more tidy picture.

VII. Discussions

FEAST has a number of signature features. By nature it works equally well regardless whether the targeted spectrum consists of dominant eigenvalues or not. It zooms in on all the targets simultaneously, at practically the same rapid convergence rate. The dimension of the subspaces, as well as the linear systems that need to be solved remained unchanged throughout a fixed targeted domain $\mathcal{C}$. Although the linear systems are of the form $z_k I - A$, they are not shifts in the familiar sense. The z_ks are not meant to be close to any eigenvalues but merely correspond to nodes of a numerical quadrature rule. Under ideal situations, they are not near any eigenvalues and none of the linear systems is ill-conditioned. Every one of these features is distinct from those associated with popular non-Hermitian eigensolvers such as unsymmetric Lanczos [39], Arnoldi [14], or Jacobi-Davidson [16], [17].

We have purposely skipped over several relevant discussions due to the limited scope of this paper. Theorems 2 through 7 focus on eigenvalues. One can show that, under appropriate assumptions, the eigenvectors obtained yield small residual and approximate the target eigenvectors. The general analysis techniques used in [23] are applicable. Nevertheless, the details would require too many lines of deltas and epsilons. We believe that the numerical experiments have provided reasonable assurances that the claims just made here are credible. The FEAST algorithms require the user to set a subspace dimension p, which should exceed the number of eigenvalues expected in the target region. In practice, this p is often chosen based on a priori knowledge or experience, or trial-and-error. A more elaborate theory exists, similar to those detailed in [23] for the Hermitian case, on estimation of the size of spectrum inside the target region $\mathcal{C}$. For example, one can use the eigenvalues of $\tilde{V}^H\tilde{U}$ ($\tilde{U}, \tilde{V}$ from Step 4 of Bi-FEAST) to estimate the eigenvalue count inside $\mathcal{C}$ and to set p. While this paper discusses exclusively the simple eigenvalue problem $Ax = \lambda x$, FEAST is designed to work on the generalized eigenvalue problem $Ax = \lambda Bx$. In short, replace the approximate spectral projector $\rho(A)$ by $\rho(A)B$, and the simple "reduced" eigenvalue problem $\tilde{A}\tilde{x} = \tilde{\lambda}\tilde{x}$ by the generalized "reduced" eigenvalue problem $\tilde{A}\tilde{x} = \tilde{\lambda}\tilde{B}\tilde{x}$ where $\tilde{B} = \tilde{U}^H B\tilde{U}$ (Step 5 and 6 of R-FEAST) or $\tilde{B} = \tilde{V}^H B\tilde{U}$ (Step 7 and 8 of Bi-FEAST). Indeed, both Laux's experiment [32] and Experiment 4 here tackled matrix pencils.

VIII. Conclusion

In the paper, we have introduced a new non-Hermitian eigensolver with rich inherent parallelism. The FEAST solver for Hermitian problems [22], [23] has been extended to non-Hermitian case in two flavors. Bi-FEAST is the "bullish-but-riskier" sibling of the more conservative R-FEAST. For well-conditioned problems, Bi-FEAST offers faster convergence of eigenvalues; R-FEAST, however, is just as fast in producing small residuals. Both are useful and complement each other.

Opportunities for further work present themselves naturally, in the directions of approximation theory, matrix analysis and parallel computing. At FEAST's core is a rational function close to 1 inside a domain $\mathcal{C}$, and 0 outside. Here we have used either a Gauss or trapezoidal quadrature rule to construct this rational function. In general, possibility abounds for other quadrature rules, either general or domain, $\mathcal{C}$, specific (see [40] for example). Alternatively, one can view this as a function approximation problem. Chebyshev polynomials [41], [42] which work well on the real line (for Hermitian problems) would not work on the complex plane in terms of approximating the $\pi(\lambda)$ function: Polynomials are analytic and must obey the maximum modulus theorem (see [43] for example). Rational approximation can contribute fruitfully here. We have already seen one such case for Hermitian problem where Zolotarev approximation is shown to outperform Gauss quadrature [44].

In non-Hermitian matrix computations, it is customary to focus on the class of diagonalizable matrices. The Cauchy integral formulation of the spectral projector always transform a Jordan block, even with nonzero superdiagonal, to either identity or zero. How well an approximate spectral projector preserves this property, and what the resulting implication on FEAST's convergence behavior in the face of deficient eigenvectors will be, are worthy pursuit that requires classical matrix and perturbation analysis.

Last but not least, FEAST offers multiple levels of parallelism: multiple target domains, multiple linear systems, with multiple right hand sides. Exploiting these parallelism fully, automatically, require much work still. On the highest level, fast partitioning of a region in the complex plane to subregions, each containing roughly the same number of eigenvalues, for the obvious sake of load balancing, is nontrivial. Challenging software engineering work is required to automatically distribute and coordinate the linear solvers – direct or iterative, sparse or dense – on multiple right hand sides, among multiple nodes, cores and threads.

As always: so much worth doing, and so little time!

References

[1] Y. Saad, "Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems," *Mathematics of Computation*, vol. 42, no. 166, pp. 567–588, 1984.

[2] ——, *Numerical Methods for Large Eigenvalue Problems.* Philadelphia: SIAM, 2011.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[3] J. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, vol. 114, no. 2, pp. 185–200, 1994.
[4] S. Odermatt, M. Luisier, and B. Witzigmann, "Bandstructure calculation using the $k \cdot p$ method for arbitrary potentials," *Journal of Applied Physics*, vol. 97, no. 4, pp. 046 104–046 104–3, 2009.
[5] D. Bindel and S. Govindjee, "Elastic PMLs for resonator anchor loss," *Internationl Journal for Numerical Methods in Engineering*, vol. 64, pp. 789–818, 2005.
[6] F. Tisseur and K. Meerbergen, "The quadratic eigenvalue problem," *SIAM Review*, vol. 43, pp. 235–286, 2001.
[7] NetLib, "Freely available software for linear algebra," 2011, `www.netlib.org/utk/people/JackDongarra/la-sw.html`.
[8] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorenson, *LAPACK Users Guide*, 3rd ed. Philadelphia: SIAM, 1999.
[9] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997.
[10] J. Demmel, *Applied Numerical Linear Algebra*. Philadelphia: SIAM, 1997.
[11] E. Polizzi, "Density-matrix-based algorithm for solving eigenvalue problems," *Physical Review B*, vol. 79, no. 115112, 2009.
[12] Z. Bai, J. Demmel, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems*. Philadelphia: SIAM, 2000.
[13] J. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*. Boston: Birkhäuser, 1985.
[14] R. Lehoucq and D. Sorensen, "Deflation techniques for an implicitly restarted arnoldi iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, pp. 789–821, 1996.
[15] B. Parlett, *The Symmetric Eigenvalue Problem*. Philadelphia: SIAM, 1998.
[16] P. Arbenz and M. E. Hochstenbach, "A Jacobi-Davidson method for solving complex symmetric eigenvalue problems," *SIAM Journal on Scientific Computing*, vol. 25, no. 5, pp. 1655–1673, 2004.
[17] G. L. G. Sleijpen and H. A. V. D. Vorst, "A Jacobi-Davidson iteration method for linear eigenvalue problems," *SIAM Review*, vol. 42, no. 2, pp. 267–293, 2000.
[18] A. Sameh and Z. Tong, "The trace minimization method for the symmetric generalized eigenvalue problem," *Journal on Computational and Applied Mathematics*, vol. 123, pp. 155–175, 2000.
[19] A. H. Sameh and J. A. Wisniewski, "A trace minimization algorithm for the generalized eigenvalue problem," *SIAM Journal on Numerical Analysis*, vol. 19, no. 6, pp. 1243–1259, 1982.
[20] A. Bai, J. Demmel, J. Dongarra, A. Petitet, H. Robinson, and K. Stanley, "The spectral decomposition of nonsymmetric matrices on distributed memory parallel computers," *SIAM Journal on Scientific Computing*, vol. 18, no. 5, pp. 1446–1461, September 1997.
[21] Z. Bai and J. Demmel, "Using the matrix sign function to compute invariant subspaces," *SIAM Journal on Matrix Analysis and Applications*, vol. 10, no. 1, pp. 205–225, January 1998.
[22] E. Polizzi, "The FEAST solver," 2009, `http://www.ecs.umass.edu/~polizzi/feast/`.
[23] P. T. P. Tang and E. Polizzi, "Subspace iteration with approximate spectral projection," 2013, `arXiv:1302.0432 [math.NA]`, submitted for publication.
[24] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.
[25] M. W. Mahoney, "Randomized algorithms for matrices and data," *Foundations and Trends in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.
[26] N. Hale, N. J. Higham, and L. N. Trefethen, "Computing a^{α}, $\log(a)$, and related matrix functions by contour integrals," *SIAM Review*, vol. 46, no. 5, pp. 2505–2523, 2008.
[27] N. J. Higham, *Functions of Matrices*. Philadelphia: SIAM, 2008.
[28] T. Sakurai and H. Sugiura, "A projection method for generalized eigenvalue problems using numerical integration," *Journal on Computational and Applied Mathematics*, vol. 159, pp. 119–128, 2003.
[29] G. W. Stewart, "Accelarating the orthogonal iteration for the eigenvalues of a hermitian matrix," *Numerische Mathematik*, vol. 13, pp. 362–376, 1969.
[30] F. L. Bauer, "On modern matrix iteration processes of Bernoulli and Graeffe types," *Journal of the Assoication of Computing Machinery*, vol. 5, pp. 246–257, 1958.
[31] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press, 1965.
[32] S. E. Laux, "Solving complex band structure problems with the FEAST eigenvalue algorithm," *Physical Review B*, vol. 86, no. 075103, 2012.
[33] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*. Boston: Academic Press, 1990.
[34] S. I. Chu, "Complex quasivibrational energy formalism for intense-field multi photon and above-threshold dissociation: Complex-scaling Fourier-grid Hamiltonian method," *Journal of Chemical Physics*, vol. 94, pp. 7901–7909, 1991.
[35] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collections," *ACM Transactions on Mathematical Software*, vol. 38, pp. 1–25, 2011.
[36] T. Kato, *Perturbation Theory for Linear Operators. Reprint of the corr. print of the 2nd ed. 1980*. Berlin: Classics in Mathematics. Springer-Verlag, 1995.
[37] A. Levin, D. Zhang, and E. Polizzi, "FEAST fundamental framework for electronic structure calculations: Reformulation and solution of the muffin-tin problem," *Computer Physics Communications*, vol. 183, pp. 2370–2375, 2012.
[38] L. Lehtovaara, V. Havu, and M. Puska, "All-electron time-dependent density functional theory with finite elements: Time-propagation approach," *Journal of Chemical Physics*, vol. 135, no. 154104, 2012.
[39] B. N. Parlett, D. R. Taylor, and Z. A. Liu, "A look-ahead Lanczos algorithm for unsymmetric matrices," *Mathematics of Computation*, vol. 44, pp. 105–124, 1985.
[40] D. Bailey and J. Borwein, "Hand-to-hand combat with thousand-digit integrals," *Journal of Computational Science*, vol. 3, pp. 77–86, 2012.
[41] Y. Zhou and Y. Saad, "A Chebyshev-Davidson algorithm for large symmetric eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 3, pp. 954–971, 2007.
[42] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, "Self-consistent-field calculations using Chebyshev-filtered subspace iteration," *Journal of Computational Physics*, vol. 219, pp. 172–184, 2006.
[43] G. Polya and G. Latta, *Complex Variables*. New York: John Wiley and Sons, Inc., 1974.
[44] G. Viaud, "The FEAST algorithm for generalised eigenvalue problems," Master's thesis, University of Oxford, Oxford, England, 2012.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)