INTERIOR EIGENSOLVER FOR SPARSE HERMITIAN DEFINITE MATRICES BASED ON ZOLOTAREV'S FUNCTIONS

YINGZHOU LI* AND HAIZHAO YANG †

Abstract. This paper proposes an efficient method for computing selected generalized eigenpairs of a sparse Hermitian definite matrix pencil (A,B). Based on Zolotarev's best rational function approximations of the signum function and conformal mapping techniques, we construct the best rational function approximation of a rectangular function supported on an arbitrary interval via function compositions with partial fraction representations. This new best rational function approximation can be applied to construct spectrum filters of (A,B) with a smaller number of poles than a direct construction without function compositions. Combining fast direct solvers and the shift-invariant generalized minimal residual method, a hybrid fast algorithm is proposed to apply spectral filters efficiently. Compared to the state-of-the-art algorithm FEAST, the proposed rational function approximation is more efficient when sparse matrix factorizations are required to solve multi-shift linear systems in the eigensolver, since the smaller number of matrix factorizations is needed in our method. The efficiency and stability of the proposed method are demonstrated by numerical examples from computational chemistry.

Keywords. Generalized eigenvalue problem; spectrum slicing; rational function approximation; sparse Hermitian matrix; Zolotarev's function; shift-invariant GMRES.

AMS subject classifications. 44A55; 65R10; 65T50

1. Introduction

Given a sparse Hermitian definite matrix pencil (A,B) (i.e., A and B are Hermitian and B is positive-definite) in $\mathbb{F}^{N\times N}$, where $\mathbb{F}=\mathbb{R}$ or $\mathbb{F}=\mathbb{C}$, and an interval (a,b) of interest, this paper aims at identifying all the eigenpairs $\{(\lambda_j,x_j)\}_{1\le j\le n_\lambda}$ [1] of (A,B) in (a,b), i.e.,

$$Ax_j=\lambda_j Bx_j \quad \text{and} \quad a<\lambda_j<b, \quad j=1,2,\ldots,n_\lambda. \tag{1.1}$$

The interior generalized eigenvalue problem not only can be applied to solve the full generalized eigenvalue problem via the spectrum slicing idea [3,10,22,24,25,27,29,31,34], but also is a stand-alone problem encountered in many fields in science and engineering (such as computational chemistry, control theory, material science, etc.), where a partial spectrum is of interest.

1.1. Related Work

A powerful tool for solving the interior generalized eigenvalue problem is the subspace iteration method accelerated by spectrum filters. Let $P_{ab}(A,B)$ be an approximate spectrum projector onto the eigen-subspace of the matrix pencil (A,B) corresponding to the desired eigenvalues in (a,b). A possible way to construct $P_{ab}(A,B)$ is to design a filter function $R_{ab}(x)$ as a good approximation to a rectangular function with a support on (a,b) (denoted as $S_{ab}(x)$), and define $P_{ab}(A,B)=R_{ab}(B^{-1}A)$. There are mainly two kinds of filter functions: polynomial filters [10, 27] and rational filters [3, 22, 24, 25, 29–31, 34]. The difficulty in designing an appropriate filter comes from the dilemma that: an accurate approximation to the spectrum projector requires

*School of Mathematical Sciences, Fudan University, (yingzhouli0417@gmail.com). https://yingzhouli.com/

†Department of Mathematics, Purdue University, (haizhao@purdue.edu). https://haizhaoyang.github.io/

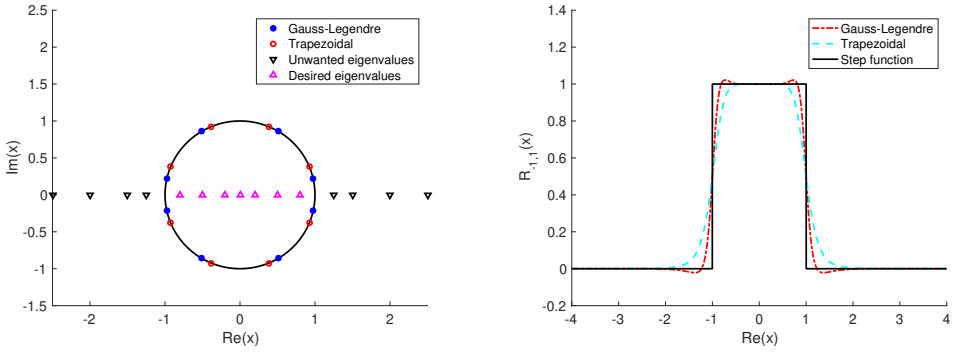
[1] Through out the paper, we assume that the exact number of eigenvalues, n_λ, is known a priori, which would simplify the presentation of the method. While, in practice, an estimated number of n_λ is enough for the algorithm.

a polynomial of high degree or a rational function with many poles; however this in turn results in expensive computational cost in applying the spectrum projector $R_{ab}(B^{-1}A)$.

In general, a rational filter can be written as follows

$$R_{ab}(x) = \alpha_0 + \sum_{j=1}^{p} \frac{\alpha_j}{x - \sigma_j}, \tag{1.2}$$

where $\{\alpha_j\}_{0\le j\le p}$ are weights, $\{\sigma_j\}_{1\le j\le p}$ are poles, and p is the number of poles. Hence, applying the spectrum projector $R_{ab}(B^{-1}A)$ to a vector v requires solving p linear systems $\{(A-\sigma_j B)^{-1}Bv\}_{1\le j\le p}$. Therefore, a large number p makes it expensive to apply the approximate spectrum projector $R_{ab}(B^{-1}A)$. A natural idea is to solve the linear systems $\{(A-\sigma_j B)^{-1}Bv\}_{1\le j\le p}$ in parallel. However, for the purposes of energy efficiency and numerical stability, an optimal p is always preferred. Extensive effort has been made to develop rational functions with p as small as possible while keeping the accuracy of the approximation.



(a) (b)

Fig. 1.1: (a) An example of a unit circle contour Γ centered at the origin, i.e., the desired spectrum range is $(-1,1)$, together with eight Gauss-Legendre quadrature points (solid blue circle) and eight Trapezoid quadrature points (red circle). The desired eigenvalues (pink up triangular) are inside the contour whereas the unwanted eigenvalues (black down triangular) are outside. (b) A rectangular function in solid black line with rational functions corresponding to the quadratures from (a).

Many rational filters in the literature were constructed by discretizing the contour integral on the complex plane,

$$\pi(x) = \frac{1}{2\pi\imath}\oint_{\Gamma} \frac{1}{x-z}\,\mathrm{d}z, \quad x \notin \Gamma \tag{1.3}$$

with an appropriate quadrature rule (e.g., the Gauss-Legendre quadrature rule [22], the trapezoidal quadrature rule [28, 34], and the Zolotarev quadrature rule [7]). Here Γ is a closed contour on the complex plane intersecting the real axis at $z=a$ and $z=b$ with all desired eigenvalues inside (a,b) and other eigenvalues outside (See Figure 1.1 (left) for an example). Suppose $\{\sigma_j\}_{1\le j\le p}$ and $\{w_j\}_{1\le j\le p}$ are the quadrature points and

weights in the discretization of the contour Γ, respectively, the contour integral (1.3) is discretized as a rational function

$$R(x)=\sum_{j=1}^{p}\frac{w_j}{2\pi\imath(x-\sigma_j)}=\alpha_0+\sum_{j=1}^{p}\frac{\alpha_j}{x-\sigma_j}, \tag{1.4}$$

where $\alpha_0=0$, and $\alpha_j=\frac{w_j}{2\pi\imath}$ for $j=1,2,\ldots,p$. Some other methods advanced with conformal maps [8,12] and optimization [29,31] can also provide good rational filters.

1.2. Contribution

Based on Zolotarev's best rational function approximations of the signum function and conformal maps, we construct the best rational function $R_{ab}(x)$ approximating a rectangular function supported on an arbitrary interval (a,b). The optimality in this paper is in terms of the uniform approximation error among the class of rational functions of the same type. Combining fast direct solvers and the shift-invariant generalized minimal residual method (GMRES), a hybrid fast algorithm is proposed to apply the spectrum filter $R_{ab}(B^{-1}A)$ to given vectors.

Suppose $a\in(a_-,a_+)$ and $b\in(b_-,b_+)$ respectively, and no eigenvalue lies in (a_-,a_+) and (b_-,b_+). The proposed rational filter $R_{ab}(x)$ is constructed via the composition of Zolotarev's functions as follows

$$R_{ab}(x)=\frac{Z_{2r}(\widehat{Z}_{2r}(T(x);\ell_1);\ell_2)+1}{2}, \tag{1.5}$$

where $Z_{2r}(x;\ell)$ is the Zolotarev's function of type $(2r-1,2r)$, $\widehat{Z}_{2r}(x;\ell)$ is the scaled Zolotarev's function

$$\widehat{Z}_{2r}(x;\ell)=\frac{Z_{2r}(x;\ell)}{\max_{x\in[\ell,1]}Z_{2r}(x;\ell)}, \tag{1.6}$$

and $T(x)$ is a Möbius transformation of the form

$$T(x)=\gamma\frac{x-\alpha}{x-\beta} \tag{1.7}$$

with $\alpha\in(a_-,a_+)$ and $\beta\in(b_-,b_+)$ such that

$$T(a_-)=-1,\quad T(a_+)=1,\quad T(b_-)=\ell_1,\quad \text{and } T(b_+)=-\ell_1. \tag{1.8}$$

In the above construction, the variables α, β, γ, ℓ_1, and ℓ_2 are all determined by a_-, a_+, b_-, and b_+.

The novelty of the proposed rational filter in (1.5) is to construct a high-order rational function for an arbitrary interval via the composition of two Zolotarev's functions and a Möbius transformation. This new construction can significantly improve the approximation accuracy for a rectangular function approximation even if r is small, as compared to other methods via a single Zolotarev's functions in [7]. Similar composition ideas have been applied to the signum function approximation (e.g., polar decomposition of matrices [16], full diagonalization of matrices [17], and the density matrix purification [15,18,20]), and the square root function approximation for accelerating Heron?s iteration [4,19]. After the completion of the investigation described in this paper the authors became aware of a work [7] that addresses a similar question about the optimal rational filter via Zolotarev's functions. The main difference is that: we

propose to construct high-order rational functions via function compositions, while [7] directly constructs the rational function without compositions. Function composition can reduce the number of direct matrix factorizations needed in the computation and hence would reduce the computational time. This idea has not been explored yet for computing interior eigenpairs and is the main contribution of our paper.

An immediate challenge arises from applying the composition of functions $R_{ab}(B^{-1}A)$ in (1.5) to given vectors when A and B are sparse matrices. Directly computing $R_{ab}(B^{-1}A)$ will destroy the sparsity of A and B since $R_{ab}(B^{-1}A)$ is dense. Fortunately, the function composition structure in (1.5) admits a hybrid fast algorithm for the matrix-vector multiplication (matvec) $R_{ab}(B^{-1}A)V$, where A and B are sparse Hermitian matrices of size N with $O(N)$ nonzero entries, B is positive definite, and V is a tall skinny matrix of size N by $O(1)$. We apply the multifrontal method [5,13] to solve the sparse linear systems involved in $\widehat{Z}_{2r}(T(B^{-1}A);\ell_1)V$. The multifrontal method consists of two parts: the factorization of sparse matrices and the application of the factorization. Once sparse factors have been constructed, evaluating $\widehat{Z}_{2r}(T(B^{-1}A);\ell_1)V$ is efficient; in this sense, the multifrontal method converts the dense matrix $\widehat{Z}_{2r}(T(B^{-1}A);\ell_1)$ into an operator with a fast application. Since the Zolotarev's function well approximates the signum function, the matrix $\widehat{Z}_{2r}(T(B^{-1}A);\ell_1)$ has a condition number close to 1. Therefore, the computation $Z_{2r}(\widehat{Z}_{2r}(T(B^{-1}A);\ell_1);\ell_2)V$ can be carried out efficiently by the GMRES iterative method. As we shall see later, by the shift-invariant property of Krylov subspace, the computational time can be further reduced in the GMRES.

When we incorporate the above hybrid fast algorithm into the subspace iteration method, the factorization time of the multifrontal method can be treated as precomputation, since all the multi-shift linear systems in every iteration remain unchanged. Since $Z_{2r}(\widehat{Z}_{2r}(T(B^{-1}A);\ell_1);\ell_2)$ is able to approximate the desired spectrum projector of (A,B) accurately, the subspace iteration method usually needs only one or two iterations to identify desired eigenpairs up to an 10^{-10} relative error. Hence, the dominant computing time in the proposed interior eigensolver is the factorization time in the multifrontal method.

1.3. Organization

In what follows, we introduce the subspace iteration, the best rational filter, and the hybrid fast algorithm in Section 2. In Section 3, extensive numerical examples of a wide range of sparse matrices are presented to demonstrate the efficiency of the proposed algorithms. Finally, we conclude this paper with a short discussion in Section 4.

2. Algorithm

First, we recall a standard subspace iteration accelerated by a rational filter for interior generalized eigenvalue problems in Section 2.1. Second, we introduce the best rational filter $R_{ab}(x)$ in (1.5) and show its efficiency of approximating the rectangular function $S_{ab}(x)$ on the interval

$$(-\infty,a_-]\cup[a_+,b_-]\cup[b_+,\infty), \tag{2.1}$$

where (a_-,a_+) and (b_-,b_+) are eigengaps around a and b, i.e., there is no eigenvalue inside these two intervals. Third, the hybrid fast algorithm for evaluating the matvec $R_{ab}(B^{-1}A)V$ is introduced in Section 2.3.

Throughout this paper, we adopt MATLAB notations for submatrices and indices. Besides usual MATLAB notations, we summarize a few notations that would be used in the rest of the paper without further explanation in Table 2.1.

Notation	Description
N	Size of the matrix
$\mathbb{F}$	Either $\mathbb{R}$ or $\mathbb{C}$
A,B	Sparse Hermitian definite matrix of size $N\times N$
(A,B)	Matrix pencil
(a,b)	Interval of interest on the spectrum of (A,B)
$(a_-,a_+),(b_-,b_+)$	Eigengaps around a and b respectively
n_λ	Number of eigenvalues in the interval
k	Oversampling constant

Table 2.1: Commonly used notations.

2.1. Subspace iteration with rational filters

Various subspace iteration methods have been proposed and analyzed in the literature. For the completeness of the presentation, we introduce a standard one in conjunction of a rational filter in Algorithm 1 for the interior generalized eigenvalue problem for a matrix pencil (A,B) on a spectrum interval (a,b).

Algorithm 1: A standard subspace iteration method

input : Sparse Hermitian matrix pencil (A,B), a spectrum range (a,b), the number of eigenpairs n_λ, and a rational filter $R_{ab}(x)$

output: A diagonal matrix Λ with diagonal entries being the eigenvalues of (A,B) on (a,b), V are the corresponding eigenvectors

Generate orthonormal random vectors $Q\in\mathbb{F}^{N\times(n_\lambda+k)}$.
while *not convergent*[a] **do**
  $Y=R_{ab}(B^{-1}A)Q$
  Compute $\tilde{A}=Y^*AY$ and $\tilde{B}=Y^*BY$
  Solve $\widetilde{A}\widetilde{Q}=\widetilde{\Lambda}\widetilde{B}\widetilde{Q}$ for $\widetilde{\Lambda}$ and $\widetilde{Q}$
  Update $Q=Y\widetilde{Q}$
end
$\mathcal{I}=\{i\,|\,a<\widetilde{\Lambda}(i,i)<b\}$
$\Lambda=\widetilde{\Lambda}(\mathcal{I},\mathcal{I})$
$V=Q(:,\mathcal{I})$

[a] Locking can be applied in the iteration.

The main cost in Algorithm 1 is to compute $Y=R_{ab}(B^{-1}A)Q$, since any other steps scale at most linearly in N or even independent of N. If the rational function $R_{ab}(x)$ is not a good approximation to the rectangular function $S_{ab}(x)$, it might take many iterations for Algorithm 1 to converge. Our goal is to get an accurate rational function approximation $R_{ab}(x)$ so that only a small number of iterations is sufficient to estimate the eigenpairs of (A,B) with machine accuracy. The method to achieve the goal will be discussed in the next two subsections.

2.2. Best rational filter by Zolotarev's functions

In what follows, we introduce basic definitions and theorems for rational function approximations. Let $\mathcal{P}_r$ denote the set of all polynomials of degree r. A rational function $R(x)$ is said to be of type (r_1,r_2) if $R(x)=\frac{P(x)}{Q(x)}$ with $P(x)\in\mathcal{P}_{r_1}$ and $Q(x)\in\mathcal{P}_{r_2}$. We

denote the set of all rational functions of type (r_1,r_2) as $\mathcal{R}_{r_1,r_2}$. For a given function $f(x)$ and a rational function $R(x)$, the approximation error in a given domain Ω is quantified by the infinity norm

$$\|f-R\|_{L^\infty(\Omega)} = \sup_{x\in\Omega} |f(x)-R(x)|. \tag{2.2}$$

A common problem in the rational function approximation is the minimax problem that identifies $R(x)\in\mathcal{R}_{r_1,r_2}$ satisfying

$$R = \arg\min_{g\in\mathcal{R}_{r_1,r_2}} \|f-g\|_{L^\infty(\Omega)}. \tag{2.3}$$

More specifically, the minimax problem of interest for matrix computation is either

$$R = \arg\min_{g\in\mathcal{R}_{2r-1,2r}} \|\mathrm{sign}(x)-g(x)\|_{L^\infty([-1,-\ell]\cup[\ell,1])}, \tag{2.4}$$

where r is a given integer and $\ell\in(0,1)$ is a given parameter, or

$$R = \arg\min_{g\in\mathcal{R}_{(2r)^2,(2r)^2}} \|S_{ab}(x)-g(x)\|_{L^\infty((-\infty,a_-)\cup(a_+,b_-)\cup(b_+,\infty))}, \tag{2.5}$$

where r is a given integer, $a_-<a$ and $a_+>a$ are two parameters around a, $b_-<b$ and $b_+>b$ are two parameters around b. The problem in (2.4) with g of the particular type $\mathcal{R}_{2r-1,2r}$, has a unique solution and the explicit expression of the solution is given by Zolotarev [35]. We denote this best rational approximation to the signum function by $Z_{2r}(x;\ell)$. To be more precise, the following theorem summarizes one of Zolotarev's conclusions which is rephrased by Akhiezer in Chapter 9 in [2], and by Petrushev and Popov in Chapter 4.3 in [21].

THEOREM 2.1 (Zolotarev's function). *The best uniform rational approximant of type $(2r,2r)$ for the signum function $sign(x)$ on the set $[-1,-\ell]\cup[\ell,1]$, $0<\ell<1$, is given by*

$$Z_{2r}(x;\ell) := Mx\frac{\prod_{j=1}^{r-1}(x^2+c_{2j})}{\prod_{j=1}^{r}(x^2+c_{2j-1})} \in \mathcal{R}_{2r-1,2r}, \tag{2.6}$$

where $M>0$ is a unique constant such that

$$\min_{x\in[-1,-\ell]} Z_{2r}(x;\ell)+1 = \min_{x\in[\ell,1]} Z_{2r}(x;\ell)-1. \tag{2.7}$$

The coefficients $c_1,c_2,\ldots,c_{2r-1}$ are given by

$$c_j = \ell^2\frac{sn^2\left(\frac{jK'}{2r};\ell'\right)}{cn^2\left(\frac{jK'}{2r};\ell'\right)}, \quad j=1,2,\ldots,2r-1, \tag{2.8}$$

where $sn(x;\ell')$ and $cn(x;\ell')$ are the Jacobi elliptic functions (see [1, 2]), $\ell'=\sqrt{1-\ell^2}$, and $K'=\int_0^{\pi/2}\frac{d\theta}{\sqrt{1-(\ell')^2\sin^2\theta}}$.

By Add. E in [1], the maximum approximation error $\delta(2r,\ell) := \|\mathrm{sign}(x)-Z_{2r}(x;\ell)\|_{L^\infty}$ is attained at $2r+1$ points $x_1:=\ell<x_2<\cdots<x_{2r}<x_{2r+1}:=1$ on the interval $[\ell,1]$ and also $2r+1$ points $x_{-j}:=-x_j$, $j=1,2,\ldots,2r+1$, on the interval $[-1,-\ell]$. The function $\mathrm{sign}(x)-Z_{2r}(x;\ell)$ equioscillates between the x_j's; in particular,

$$1-Z_{2r}(x_j;\ell) = (-1)^{j+1}\delta(2r,\ell), \quad j=1,2,\ldots,2r+1. \tag{2.9}$$

The approximation error of Zolotarev's functions as an approximant to $\mathrm{sign}(x)$ decreases exponentially with degree $2r$ ([21] Section 4.3), i.e.

$$\delta(2r,\ell)\approx C\rho^{-2r} \tag{2.10}$$

for some positive C and $\rho>1$ that depends on ℓ. In more particular, Gončar [6] gave the following quantitative estimation on the approximation error, $\delta(2r,\ell)$:

$$\frac{2}{\rho^{2r}+1}\leq\delta(2r,\ell)\leq\frac{2}{\rho^{2r}-1}, \tag{2.11}$$

where

$$\rho=\exp\left(\frac{\pi K(\mu')}{4K(\mu)}\right), \tag{2.12}$$

$\mu=\frac{1-\sqrt{\ell}}{1+\sqrt{\ell}}$, $\mu'=\sqrt{1-\mu^2}$, and $K(\mu)=\int_0^{\pi/2}\frac{\mathrm{d}\theta}{\sqrt{1-(\mu)^2\sin^2\theta}}$ is the complete elliptic integral of the first kind for the modulus μ.

Even though the approximation error $\delta(2r,\ell)$ decreases exponentially in r, the decay rate of $\delta(2r,\ell)$ in r might still be slow if ρ is small. In fact, ρ could be small in many applications when eigengaps are small. As we shall see later, if the eigenvalues cluster together, ℓ should be very small and hence ρ is small by (2.12). As we have discussed earlier in the introduction of this paper, it is not practical to use a large r due to the computational expense and numerical instability. This motivates the study of the composition of Zolotarev's functions in $\mathcal{R}_{2r-1,2r}$, which constructs a high order Zolotarev's function in $\mathcal{R}_{(2r)^2-1,(2r)^2}$. Such a composition has a much smaller approximation error

$$\delta(4r^2,\ell)\approx C\rho^{-4r^2}. \tag{2.13}$$

For simplicity, let us use the rescaled Zolotarev's function defined by

$$\widehat{Z}_{2r}(x;\ell)=\frac{Z_{2r}(x;\ell)}{\max_{x\in[\ell,1]}Z_{2r}(x;\ell)}. \tag{2.14}$$

Note that $\max_{x\in[\ell,1]}\widehat{Z}_{2r}(x;\ell)=1$, and $\widehat{Z}_{2r}(x;\ell)$ maps the set $[-1,-\ell]\cup[\ell,1]$ onto $[-1,-\widehat{Z}_{2r}(\ell;\ell)]\cup[\widehat{Z}_{2r}(\ell;\ell),1]$. Hence, if one defines a composition via

$$S(x;\ell_1)=Z_{2r}(\widehat{Z}_{2r}(x;\ell_1);\ell_2), \tag{2.15}$$

where $\ell_2=\widehat{Z}_{2r}(\ell_1;\ell_1)$, then $S(x;\ell_1)\in\mathcal{R}_{(2r)^2-1,(2r)^2}$ is the best uniform rational approximant of type $((2r)^2,(2r)^2)$ for the signum function $\mathrm{sign}(x)$ on the set $[-1,-\ell_1]\cup[\ell_1,1]$. This optimal approximation is an immediate result of a more general theorem as follows.

THEOREM 2.2. *Let $\widehat{Z}_{2r_1}(x;\ell_1)\in\mathcal{R}_{2r_1-1,2r_1}$ be the rescaled Zolotarev's function corresponding to $\ell_1\in(0,1)$, and $Z_{2r_2}(x;\ell_2)\in\mathcal{R}_{2r_2-1,2r_2}$ be the Zolotarev's function corresponding to $\ell_2:=\widehat{Z}_{2r_1}(\ell_1;\ell_1)$. Then*

$$Z_{2r_2}(\widehat{Z}_{2r_1}(x;\ell_1);\ell_2)=Z_{(2r_1)(2r_2)}(x;\ell_1). \tag{2.16}$$

The proof of Theorem 2.2 is similar to Theorem 3 in [17]. Hence, we leave it to readers.

Finally, given a desired interval (a,b) and the corresponding eigengaps, (a_-,a_+) and (b_-,b_+), to answer the best rational function approximation in (2.5), we construct a uniform rational approximant $R_{ab}(x)\in\mathcal{R}_{(2r)^2,(2r)^2}$ via the Möbius transformation $T(x)$ as follows

$$R_{ab}(x)=\frac{S(T(x);\ell_1)+1}{2}=\frac{Z_{2r}(\widehat{Z}_{2r}(T(x);\ell_1);\ell_2)+1}{2}, \tag{2.17}$$

where $\ell_2=\widehat{Z}_{2r}(\ell_1;\ell_1)$ and

$$T(x)=\gamma\frac{x-\alpha}{x-\beta} \tag{2.18}$$

with $\alpha\in(a_-,a_+)$ and $\beta\in(b_-,b_+)$ such that

$$T(a_-)=-1,\quad T(a_+)=1,\quad T(b_-)=\ell_1,\quad \text{and } T(b_+)=-\ell_1. \tag{2.19}$$

We would like to emphasize that the variables α, β, γ, ℓ_1, and ℓ_2 are determined by a_-, a_+, b_-, and b_+ via solving the equations in (2.19) in the above construction. In practice, a_-, a_+, b_-, and b_+ can be easily calculated from a and b. We fixed the buffer region $(a_-,a_+)\cup(b_-,b_+)$ first according to the eigengaps of target matrices and construct a Möbius transformation adaptive to this region. This adaptive idea is natural but does not seem to have been considered before in the literature. Following Corollary 4.2 in [7] on can easily prove that when $a_-=-\frac{1}{c}$, $a_+=-c$, $b_-=c$, and $b_+=\frac{1}{c}$ for some $c>0$, the rational function in (2.17) is the best rational function approximation to the step function $S_{-cc}(x)$ among all the rational functions in $\{R(T(x)):R(x)\in\mathcal{R}_{(2r)^2-1,(2r)^2}\}\subset\mathcal{R}_{(2r)^2,(2r)^2}$, where $T(x)$ is the Möbius transformation satisfying (2.19). The following theorem shows that $R_{ab}(x)$ in (2.17) is indeed the best uniform rational approximant of type $((2r)^2,(2r)^2)$ for more general a_-, a_+, b_-, and b_+ among a larger class of rational functions. [7] proved a similar theorem very briefly and our proof of Theorem 2.3 is different to that of [7]. The main purpose of our proof below is to make the paper self-contained.

THEOREM 2.3. *The rational function $R_{ab}(x)$ given in (2.17) satisfies the following properties:*

1) *$R_{ab}(x)$ is the best uniform rational approximant of type $((2r)^2,(2r)^2)$ of the rectangular function $S_{ab}(x)$ on*

$$\Omega=(-\infty,a_-]\cup[a_+,b_-]\cup[b_+,\infty), \tag{2.20}$$

where (a_-,a_+) and (b_-,b_+) are eigengaps.

2) *The error curve $e(x):=S_{ab}(x)-R_{ab}(x)$ equioscillates on Ω with the maximal error*

$$\delta_0:=\max_{x\in\Omega}|e(x)|=\min_{g\in\mathcal{R}_{(2r)^2,(2r)^2}}\|S_{ab}(x)-g(x)\|_{L^\infty(\Omega)} \tag{2.21}$$

and

$$\frac{2}{\rho^{(2r)^2}+1}\le\delta_0\le\frac{2}{\rho^{(2r)^2}-1},\quad \rho=\rho(\ell_1)>1, \tag{2.22}$$

where

$$\rho(\ell_1)=\exp\left(\frac{\pi K(\mu')}{4K(\mu)}\right),$$

$\mu=\frac{1-\sqrt{\ell_1}}{1+\sqrt{\ell_1}}$, $\mu'=\sqrt{1-\mu^2}$, and $K(\mu)=\int_0^{\pi/2}\frac{d\theta}{\sqrt{1-\mu^2\sin^2\theta}}$ is the complete elliptic integral of the first kind for the modulus μ.

Proof.

Note that inserting a rational transformation of type $(1,1)$ into a rational function of type $((2r)^2-1,(2r)^2)$ results in a rational function of type $((2r)^2,(2r)^2)$. Since $S(x;\ell_1)\in\mathcal{R}_{(2r)^2-1,(2r)^2}$ and Möbius transform $T(x)\in\mathcal{R}_{1,1}$, we know $R_{ab}(x)\in\mathcal{R}_{(2r)^2,(2r)^2}$. In the following proof, we will first show that $R_{ab}(x)$ is the best uniform rational approximant of type $((2r)^2,(2r)^2)$ to the rectangular function on Ω and then derive the error estimator.

Suppose $R_{ab}(x)$ is not the best uniform rational approximant of type $((2r)^2,(2r)^2)$ of the rectangular function $S_{ab}(x)$ on

$$\Omega=(-\infty,a_-]\cup[a_+,b_-]\cup[b_+,\infty), \tag{2.23}$$

then there exists another rational function $\tilde{R}(x)$ in $\mathcal{R}_{(2r)^2,(2r)^2}$ such that

$$\left\|S_{ab}(x)-\tilde{R}(x)\right\|_{L^\infty(\Omega)}<\|S_{ab}(x)-R_{ab}(x)\|_{L^\infty(\Omega)}.$$

Let $T^{-1}(x)$ denote the inverse transform of the Möbius transformation $T(x)$ in (2.18), and we have $T^{-1}\in\mathcal{R}_{1,1}$. Note that inserting a rational transformation of type $(1,1)$ into a rational function of type $((2r)^2,(2r)^2)$ results in a rational function of type $((2r)^2,(2r)^2)$. Hence, $2\tilde{R}(T^{-1}(x))-1$ is a rational function approximant in $\mathcal{R}_{(2r)^2,(2r)^2}$ of the signum function $\operatorname{sign}(x)$ on the set $[-1,-\ell_1]\cup[\ell_1,1]$. Note that the Möbius transformations $T(x)$ and $T^{-1}(x)$ are bijective maps that do not change the approximation errors, we have

$$\begin{aligned}&\left\|\operatorname{sign}(x)-(2\tilde{R}(T^{-1}(x))-1)\right\|_{L^\infty([-1,-\ell_1]\cup[\ell_1,1])}=2\left\|S_{ab}(x)-\tilde{R}(x)\right\|_{L^\infty(\Omega)}\\<&\,2\|S_{ab}(x)-R_{ab}(x)\|_{L^\infty(\Omega)}=\|\operatorname{sign}(x)-S(x;\ell_1)\|_{L^\infty([-1,-\ell_1]\cup[\ell_1,1])}.\end{aligned}$$

The inequality

$$\left\|\operatorname{sign}(x)-(2\tilde{R}(T^{-1}(x))-1)\right\|_{L^\infty([-1,-\ell_1]\cup[\ell_1,1])}<\|\operatorname{sign}(x)-S(x;\ell_1)\|_{L^\infty([-1,-\ell_1]\cup[\ell_1,1])}$$

conflicts with the fact that $S(x;\ell_1)\in\mathcal{R}_{(2r)^2-1,(2r)^2}$ is the best rational approximant (among all rational functions of type $((2r)^2,(2r)^2)$) of the signum function on $[-1,-\ell_1]\cup[\ell_1,1]$ by (2.15) and (2.16). Hence, our previous assumption that $R_{ab}(x)$ is not the best uniform rational approximant of type $((2r)^2,(2r)^2)$ of the rectangular function $S_{ab}(x)$ on Ω is false. This proves the first statement of Theorem 2.3.

The error inequalities in Property 2) follow from Gončar's quantitative estimation on the approximation error of Zolotarev's functions in [6] and the bijective transformation in (2.18).

□

An immediate result of Theorem 2.3 is

$$\delta_0=\|S_{ab}(x)-R(x)\|_{L^\infty(\Omega)}=C_{4r^2}\rho^{-4r^2}, \tag{2.24}$$

with $1\le\frac{2}{1+\rho^{-(2r)^2}}\le C_{4r^2}\le\frac{2}{1-\rho^{-(2r)^2}}$.

To illustrate this improvement, we compare the performance of the proposed rational filter in (2.17) with other existing rational filters that are constructed by discretizing the complex value contour integral

$$\pi(x)=\frac{1}{2\pi i}\oint_\Gamma\frac{1}{x-z}\,\mathrm{d}z,\quad x\notin\Gamma \tag{2.25}$$

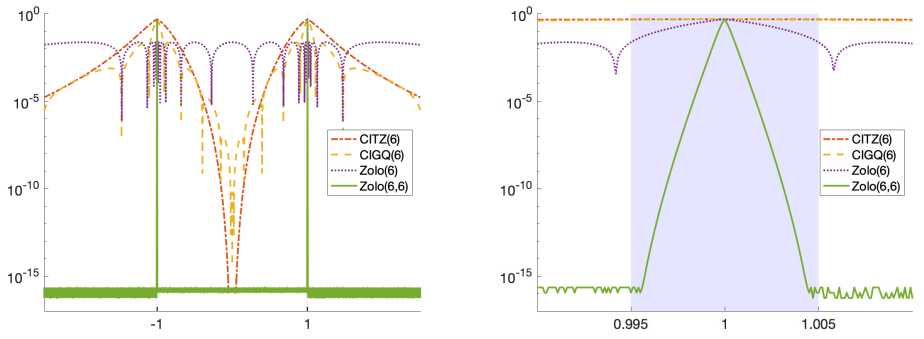


Fig. 2.1: This figure shows the approximation error of various rational filters as an approximation of the rectangular function supported on $(-1,1)$. The eigengaps around 1 and -1 is set to be 10^{-2}. These functions include: the trapezoidal filter [28,34] (denoted as CITZ(r), where r is the number of poles), the Gauss-Legendre filter [22] (denoted as CIGQ(r), where r is the number of poles), the Zolotarev approximation (denoted as Zolo(r), where r is the degree), and the proposed Zolotarev filter via compositions (denoted as Zolo(r,r), where r is the degree). (a) shows the approximation on $[-2.5,2.5]$ and (b) zooms in on $[0.99,1.01]$. Light purple areas are the buffer areas in which it is not necessary to condiser the approximation accuracy because of the eigengaps.

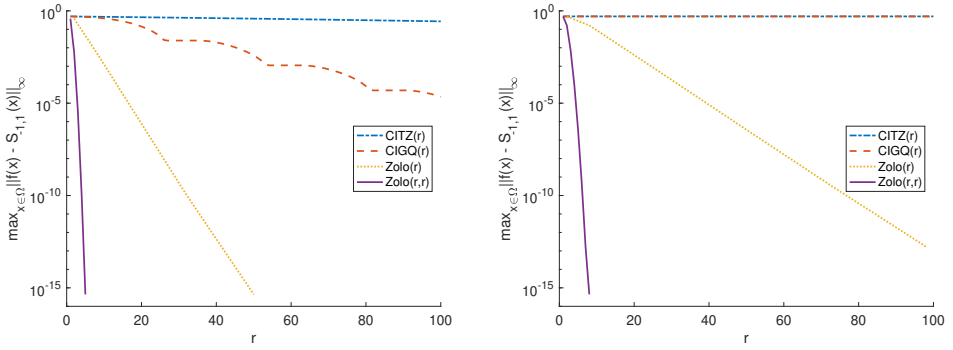


Fig. 2.2: This figure shows the approximation error against degree r for various rational functions as an approximation of the rectangular function supported on $(-1,1)$. The eigengaps around -1 and 1 are set to be 10^{-2} in (a) and 10^{-6} in (b). The approximation errors of Zolo(r,r) decay significantly faster than other methods. In (b), the line for CITZ is overwritten by that of CIGQ.

with an appropriate quadrature rule (e.g., the Gauss-Legendre quadrature rule [22] and the trapezoidal quadrature rule [28, 34]). Since the dominant cost of applying all these filters is the sparse matrix factorization, we fix the number of matrices to be factorized and compare the approximation error of various filters. The results in Figure 2.1 verifies the advantage of the proposed rational filter over existing rational filters and shows that 6 matrix factorizations are enough to construct the composition of Zolotarev's rational function approximating a rectangular function within a machine accuracy. Here the eigengaps are 10^{-2}. Figure 2.2 further explores the decay for the errors in L^∞ norm for different methods. Figure 2.2a is the decay property for problem with eigengaps 10^{-2} whereas Figure 2.2b shows the decay property for problem with eigengaps 10^{-6}.

2.3. A hybrid algorithm for applying the best rational filter

In this section, we introduce a hybrid algorithm for applying the best rational filter $R_{ab}(x)$ constructed in Section 2.2, i.e., computing the matvec $R_{ab}(B^{-1}A)V$ when A and B are sparse Hermitian matrices in $\mathbb{F}^{N\times N}$ and V is a vector in $\mathbb{F}^N$. Recall that the rational filter $R_{ab}(x)$ is constructed by

$$R_{ab}(x) = \frac{Z_{2r}(\widehat{Z}_{2r}(T(x);\ell_1);\ell_2)+1}{2}. \tag{2.26}$$

Hence, it is sufficient to show how to compute $Z_{2r}(\widehat{Z}_{2r}(T(B^{-1}A);\ell_1);\ell_2)V$ efficiently.

For the sake of numerical stability and parallel computing, a rational function is usually evaluated via a partial fraction representation in terms of a sum of fractions involving polynomials of low degree. For the Zolotarev's function $Z_{2r}(x;\ell)$ introduced in (2.6), we have the following partial fraction representation[2]. The reader is referred to Appendix for the proof.

PROPOSITION 2.1. *The function $Z_{2r}(x;\ell)$ as in (2.6) can be reformulated as*

$$Z_{2r}(x;\ell) = Mx\frac{\prod_{j=1}^{r-1}(x^2+c_{2j})}{\prod_{j=1}^{r}(x^2+c_{2j-1})} = Mx\left(\sum_{j=1}^{r}\frac{a_j}{x^2+c_{2j-1}}\right), \tag{2.27}$$

where

$$a_j = \frac{b_j}{c_{2r-1}-c_{2j-1}} \tag{2.28}$$

for $j=1,\ldots,r-1$, and

$$a_r = 1-\sum_{j=1}^{r-1}\frac{b_j}{c_{2r-1}-c_{2j-1}}. \tag{2.29}$$

Here

$$b_j = (c_{2j}-c_{2j-1})\prod_{k=1,k\neq j}^{r-1}\frac{c_{2k}-c_{2j-1}}{c_{2k-1}-c_{2j-1}} \tag{2.30}$$

for $j=1,\ldots,r-1$, $\{c_j\}$ and M are given in (2.8).

[2]The existence of the partial fraction representation is well-known. We present our formulas for the representation for the purpose of making our algorithm easier to implement for researchers who are interested in our work.

If complex coefficients are allowed, the following corollary can be derived from Proposition 2.1 directly.

COROLLARY 2.1. *The function $Z_{2r}(x;\ell)$ as in (2.6) can be reformulated as*

$$Z_{2r}(x;\ell) = \frac{M}{2}\sum_{j=1}^{r}\left(\frac{a_j}{x+\imath\sqrt{c_{2j-1}}}+\frac{a_j}{x-\imath\sqrt{c_{2j-1}}}\right), \tag{2.31}$$

where a_j and c_{2j-1} are as defined in Proposition 2.1.

By Proposition 2.1, we obtain the partial fraction representation of $Z_{2r}(T(x);\ell)$ as follows, where $T(x)$ is a Möbius transformation $T(x)=\gamma\frac{x-\alpha}{x-\beta}$. The reader is referred to Appendix for the proof.

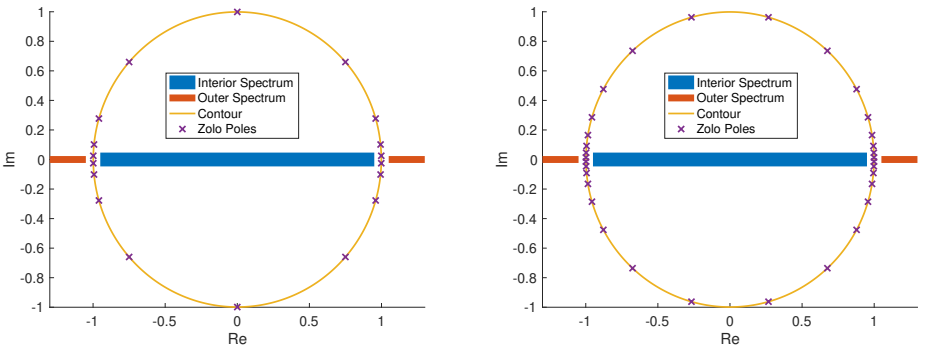
PROPOSITION 2.2. *The function $Z_{2r}(T(x);\ell)$ can be reformulated as*

$$Z_{2r}(T(x);\ell) = M\sum_{j=1}^{r}\frac{a_j\gamma}{\gamma^2+c_{2j-1}}+M\sum_{j=1}^{r}\left(\frac{w_j}{x-\sigma_j}+\frac{\bar{w}_j}{x-\bar{\sigma}_j}\right). \tag{2.32}$$

where

$$\sigma_j = \frac{\gamma\alpha+\imath\sqrt{c_{2j-1}}\beta}{\gamma+\imath\sqrt{c_{2j-1}}}, \quad w_j = \frac{a_j(\sigma_j-\beta)}{2(\gamma+\imath\sqrt{c_{2j-1}})}. \tag{2.33}$$

REMARK 2.1. *In the rest of this paper, we denote the constants associated with $Z_{2r}(x;\ell_2)$ as a_j, c_{2j-1}, σ_j, and w_j for $j=1,\ldots,r$; and the constants associated with $\widehat{Z}_{2r}(x;\ell_1)$ as $\widehat{a}_j, \widehat{c}_{2j-1}, \widehat{\sigma}_j$, and $\widehat{w}_j$ for $j=1,\ldots,r$.*



(a) (b)

Fig. 2.3: The corresponding contour integral discretization of Zolotarev's function composed with Möbius transformation. The eigengaps here are $(-1.1,-0.9)$ and $(0.9,1.1)$ and the contour is a circle centered at origin with radius 0.998749. The discretization points are calculated as Proposition 2.2. (a) $r=9$ provides 18 poles; (b) $r=16$ provides 32 poles.

Proposition 2.2 can be viewed as a discretization of a contour at poles, σ_j and $\widehat{\sigma}_j$ with weights w_j and $\widehat{w}_j$ for $j=1,\ldots,r$. The contour is a circle centered on the real axis cutting through the eigengaps. Figure 2.3 demonstrate an example with eigengaps

$(-1.1,-0.9)$ and $(0.9,1.1)$. The calculated contour is a circle centered at origin with radius 0.998749. Meanwhile, the pole locations and the corresponding weights are provided by Proposition 2.2. Figure 2.3a adopts $r=9$ which is also the composition of two Zolotarev's functions with degree 3 whereas Figure 2.3b adopts $r=16$ which is also the composition of two Zolotarev's functions with degree 4.

With these propositions ready, we now introduce the hybrid algorithm for computing the matvec $R_{ab}(B^{-1}A)V = Z_{2r}(\widehat{Z}_{2r}(T(B^{-1}A);\ell_1);\ell_2)V$. This hybrid algorithm consists of two parts of linear system solvers: an inner part and an outer part. The inner part implicitly computes the matvec $\widehat{Z}_{2r}(T(B^{-1}A);\ell_1)V := GV$ via fast direct solvers. Once GV has been implicitly computed, the matrix G can be viewed as an operator with fast application algorithm, where each application costs nearly $O(N)$ operations in many applications. The outer part computes $Z_{2r}(G;\ell_2)V$ using a GMRES method when the fast matvec GV is available. Since the matrix G has singular values greater than $\ell_2 = \widehat{Z}_{2r}(\ell_1;\ell_1)$, which is a number close to 1, a few steps of iterations in GMRES method are enough to solve the linear systems in the matvec $Z_{2r}(G;\ell_2)V$ accurately. In practice, the iteration number varies from 6 to 25.

In particular, by Proposition 2.2,

$$\begin{aligned} GV =& \widehat{Z}_{2r}(T(B^{-1}A);\ell_1)V \\ =& \widehat{M}\sum_{j=1}^{r}\frac{\widehat{a}_j\gamma}{\gamma^2+\widehat{c}_{2j-1}}V + \widehat{M}\sum_{j=1}^{r}\left(\left(\widehat{w}_j\left(A-\widehat{\sigma}_jB\right)^{-1}BV\right)+\left(\overline{\widehat{w}}_j\left(A-\overline{\widehat{\sigma}}_jB\right)^{-1}BV\right)\right) \\ =& \widehat{M}\sum_{j=1}^{r}\frac{\widehat{a}_j\gamma}{\gamma^2+\widehat{c}_{2j-1}}V + \widehat{M}\sum_{j=1}^{r}\left(\left(\widehat{w}_j\left(A-\widehat{\sigma}_jB\right)^{-1}BV\right)+\left(\overline{\widehat{w}}_j\left(A-\widehat{\sigma}_jB\right)^{-*}BV\right)\right). \end{aligned} \tag{2.34}$$

The third equality holds since A and B are Hermitian matrices. Hence, evaluating $\widehat{Z}_{2r}(T(B^{-1}A);\ell_1)V$ boils down to solving r linear systems of the form

$$(A-\widehat{\sigma}_jB)x = y \tag{2.35}$$

for $j=1,\ldots,r$. This is a set of r sparse linear systems. Since the operator G is involved in an outer function, where it is repeatedly applied, a fast and efficient algorithm for applying G is necessary. This can also be rephrased as "a fast and efficient algorithm for solving (2.35) is necessary". There are two groups of efficient algorithms for solving (2.35): direct solvers and iterative solvers with efficient preconditioners.

Fast direct solvers for sparse linear system as $A-\widehat{\sigma}_jB$ usually contains two phases. The first phase (termed as the pre-factorization phase) factorizes the sparse matrix into a product of a sequence of lower and upper triangular sparse matrices. The second phase (termed as the solving phase) solves the sequence of triangular sparse matrices efficiently against vectors. The computational complexities for both the pre-factorization and the solving phase vary from method to method, also heavily rely on the sparsity pattern of the matrix. For simplicity, we denoted the computational complexity for the pre-factorization and the solving phase as F_N and S_N respectively for matrices of size $N\times N$. Usually, F_N is of higher order in N than S_N. Particularly, we adopt the multifrontal method (MF) [5, 13] as the general direct sparse solver for all numerical examples in this paper. For sparse matrices of size $N\times N$ from two-dimensional PDEs, the computational complexities for MF are $F_N = O(N^{3/2})$ and $S_N = O(N\log N)$. While, for three-dimensional problems, MF requires $F_N = O(N^2)$ and $S_N = O(N^{4/3})$ operations.

Iterative solvers with efficient preconditioners is another efficient way to solve sparse linear systems. The construction of preconditioners is the pre-computation phase whereas the iteration together with applying the preconditioner is the solving phase. Similarly to the direct solver, the choices of iterative solvers and preconditioners highly depend on sparse matrices. For elliptic PDEs, GMRES could be used as the iterative solver for $A-\widehat{\sigma}_j B$, and MF with reduced frontals [9,11,26,33] could provide good preconditioners.

Once the fast application of G is available, we apply the classical GMRES together with the shift-invariant property of the Krylov subspace (See [23] Section 7.3) to evaluate $R_{ab}(B^{-1}A)V = Z_{2r}(\widehat{Z}_{2r}(T(B^{-1}A);\ell_1);\ell_2)V = Z_{2r}(G;\ell_2)V$. In more particular, by Corollary 2.1, we have

$$Z_{2r}(G;\ell_2)V = M\sum_{j=1}^{r}\frac{a_j}{2}\left(\left(G+\imath\sqrt{c_{2j-1}}I\right)^{-1}V+\left(G-\imath\sqrt{c_{2j-1}}I\right)^{-1}V\right), \tag{2.36}$$

where I is the identify matrix. Hence, to evaluate $Z_{2r}(G;\ell_2)V$, we need to solve multi-shift linear systems of the form

$$(G\pm\imath\sqrt{c_{2j-1}}I)x = y \tag{2.37}$$

with $2r$ shifts $\pm\imath\sqrt{c_{2j-1}}$ for $j=1,\ldots,r$. These systems are solved by the multi-shift GMRES method efficiently. In each iteration, only a single evaluation of GV is needed for all shifts. Meanwhile, since G has a condition number close to 1, only a few iterations are sufficient to solve the multi-shift systems to a high accuracy. Let the number of columns in V be $O(n_\lambda)$ and the number of iterations be m. The complexity for evaluating the rational filter $R_{ab}(B^{-1}A)V$ is $O(mn_\lambda S_N)$.

Algorithm 2: A hybrid algorithm for the rational filter $R_{ab}(B^{-1}A)$

input : A sparse Hermitian definite matrix pencil (A,B), a spectrum range (a,b), vectors V, tolerance ϵ

output: $R_{ab}(B^{-1}A)V$ as defined in (2.26)

1 Estimate eigengaps (a_-,a_+) and (b_-,b_+) for a and b respectively.

2 Solve (2.19) for ℓ_1 and Möbius transformation parameter γ,α,β.

3 Given ϵ, find the smallest order of Zolotarev's functions, r, such that our rational function approximates S_{ab} within the target accuracy ϵ by gradually increasing r.

4 Calculate function coefficients, $\widehat{M},\widehat{a}_j,\widehat{w}_j,\widehat{\sigma}_j,\widehat{c}_{2j-1}$ and ℓ_2,M,a_j,c_{2j-1} for $j=1,\ldots,r$.

5 **for** $j=1,2,\ldots,r$ **do**

6 Pre-factorize $A-\widehat{\sigma}_j B$ as K_j

7 **end**

8 Generate algorithm for operator

$$GV = \widehat{M}\sum_{j=1}^{r}\frac{\widehat{a}_j\gamma}{\gamma^2+\widehat{c}_{2j-1}}V+\widehat{M}\sum_{j=1}^{r}\left(\widehat{w}_j K_j^{-1}BV+\overline{\widehat{w}}_j K_j^{-*}BV\right).$$

9 Apply the multi-shift GMRES method for solving linear systems $(G\pm\imath\sqrt{c_{2j-1}}I)^{-1}V$ with $j=1,\ldots,r$.

10 $R_{ab}(B^{-1}A)V = \frac{M}{2}\sum_{j=1}^{r}\frac{a_j}{2}\left(\left(G+\imath\sqrt{c_{2j-1}}I\right)^{-1}V+\left(G-\imath\sqrt{c_{2j-1}}I\right)^{-1}V\right)+\frac{1}{2}V$

Algorithm 2 summarizes the hybrid algorithm introduced above for applying the rational filter $R_{ab}(B^{-1}A)$ in (2.26) to given vectors V. By taking Line 1-8 in Algorithm 2 as precomputation and inserting Line 9-10 in Algorithm 2 into Line 3 in Algorithm 1, we obtain a complete algorithm for solving the interior generalized eigenvalue problem on a given interval (a,b). When the matrix pencil (A,B) consists of sparse complex Hermitian definite matrices, the dominant cost of the algorithm is the pre-factorization of r matrices in (2.35) or Line 6 in Algorithm 2.

REMARK 2.2. *Given a desired accuracy ϵ and the parameter ℓ_1 computed from the estimated eigengaps, we can estimate the order of Zolotarev's functions efficiently, which corresponds to the third line in Algorithm 2. Notice that the L^∞ error of $S(x;\ell_1)$ as in (2.15) approximating the signum function is achieved at $x=\ell_1$. Therefore, in practice, we evaluate $S(\ell_1;\ell_1)$ for a sequence of rs and choose the smallest r such that the error is bounded by ϵ. Since the evaluation of $S(\ell_1;\ell_1)$ does not involve any matrix, the estimation of the order r can be done efficiently. If different r_1 and r_2 are of interest, a small table of $S(\ell_1;\ell_1)$ can be computed as a reference for uses to select a pair of (r_1,r_2) from it.*

3. Numerical examples

In this section, we will illustrate three examples based on different collections of sparse matrices. The first example aims to show the scaling of the proposed method; the second example shows the comparison with the state of the art algorithm for spectrum slicing problem, FEAST [7, 22]; the last example shows the efficiency of the proposed method for various kinds of sparse matrices. All numerical examples are performed on a desktop with Intel Core i7-3770K 3.5 GHz, 32 GB of memory. The proposed algorithm in this paper is implemented in MATLAB R2017b, which is shorten as "ZoloEig" or "Zolo" in this section. And the FEAST v3.0 compiled with Intel compiler produces the results in the part of "FEAST". To make the numerical results reproducible, the codes for the numerical examples can be found in the authors' personal homepages.

Throughout the numerical section, a relative error without knowing the underlying ground true eigenpairs is used to measure the accuracy of both ZoloEig and FEAST. The relative error of the estimated interior eigenpairs in the interval (a,b) is defined as

$$e_{\Lambda,X} = \max_{1\le i\le k} \frac{\|AX_i - BX_i\lambda_i\|_2}{\|\max(|a|,|b|)BX_i\|_2}, \tag{3.1}$$

where (A,B) is the matrix pencil of size N by N; $\Lambda\in\mathbb{R}^{k\times k}$ is a diagonal matrix with diagonal entries being the estimated eigenvalues in the given interval, $\{\lambda_1,\lambda_2,\ldots,\lambda_k\}$; and $X_i\in\mathbb{F}^{N\times 1}$ denotes the i-th eigenvector for $1\le i\le k$. This relative error of the eigenvalue decomposition is also used in ZoloEig as the stopping criteria. Besides the error measurement, we also define a measurement of the difficulty of the problem as the relative eigengap,

$$\delta_\lambda = \frac{\min(a_+ - a_-, b_+ - b_-)}{b_- - a_+}. \tag{3.2}$$

Such a relative eigengap, δ_λ can measure the intrinsic difficulty of the spectrum slicing problem for all existing algorithms based on polynomial filters and rational function filters.

Other notations are listed in Table 3.1. The total number of linear system solves in ZoloEig can be calculated as,

$$n_{solv} = r\cdot n_{ss}\cdot n_{iter}\cdot n_{gmres}, \tag{3.3}$$

Notation	Description
n_{ss}	Size of subspace used in ZoloEig or FEAST.
n_{iter}	Number of subspace iterations used by ZoloEig or FEAST.
n_{gmres}	Number of GMRES iterations used by ZoloEig.
n_{solv}	Total number of linear system solves used by ZoloEig or FEAST.
T_{fact}	Total factorization time used by ZoloEig in second.
T_{iter}	Total iteration time used by ZoloEig in second.
T_{total}	Total runtime used by ZoloEig in second.

Table 3.1: Notations used in numerical results.

whereas the one in FEAST is

$$n_{solv} = r \cdot n_{ss} \cdot n_{iter}. \tag{3.4}$$

3.1. Spectrum of Hamiltonian Operators

The first example is a three-dimensional Hamiltonian operator,

$$H = -\frac{1}{2}\Delta + V, \tag{3.5}$$

on $[0,1)^3$ with a Dirichlet boundary condition. Here V is a three-dimensional potential field containing three Gaussian wells with random depths uniformly chosen from $(0,1]$ and fixed radius 0.2. Figure 3.1 shows the isosurface of an instance of the 3D random Gaussian well. This example serves the role of illustrating the efficiency and complexity of the proposed new algorithm. We first discretize the domain $[0,1)^3$ by a uniform grid with n points on each dimension and the operator is discretized with 7-point stencil finite difference method that results in a sparse matrix. The multifrontal method is naturally designed for inverting such sparse matrices. In this section, we adopt Matlab "eigs" function to evaluate the smallest 88 eigenpairs as the reference. Due to the randomness in the potential, the relative eigengap of the smallest 88 eigenvalues varies a lot. In order to obtain the scaling of the algorithm, we prefer to have problems of different sizes but with similar difficulty. Therefore, we generate random potential fields until the problem has a relative eigengap between 10^{-3} and 10^{-4}. In such cases, the claimed complexity of the ZoloEig algorithm can be rigorously verified for the discretized operator of (3.5). In this example, the tolerance is set to be 10^{-8}, r is set as $(4,4)$ for all matrices, and subspaces with dimension 89 are used to recover the 88 eigenpairs.

N	δ_λ	r	$e_{\Lambda,X}$	n_{ss}	n_{iter}	n_{gmres}	n_{solv}	T_{fact}	T_{iter}	T_{total}
1728	8.6e-04	(4,4)	3.3e-15	89	1	14	4984	2.8e-01	2.2e+00	2.6e+00
8000	5.1e-04	(4,4)	3.4e-15	89	1	16	5625	1.6e+00	1.4e+01	1.6e+01
21952	4.3e-04	(4,4)	2.5e-14	89	1	15	5340	7.9e+00	4.5e+01	5.2e+01
46656	6.6e-04	(4,4)	3.7e-13	89	1	15	5340	2.9e+01	1.1e+02	1.4e+02
85184	2.4e-04	(4,4)	2.5e-12	89	1	16	5696	8.3e+01	2.4e+02	3.3e+02
140608	1.6e-04	(4,4)	4.2e-14	89	1	17	6052	2.2e+02	4.9e+02	7.1e+02

Table 3.2: Numerical results for 3D Hamiltonian Operators. N is the size of the sparse matrix, r is the order used in ZoloEig, other notations are as defined in Table 3.1.

Figure 3.2 shows the running time and the relative error of eigenvalues, $e_{\Lambda,X}$. The 3D problem size varies from 12^3 to 52^3 and the corresponding matrix size varies from

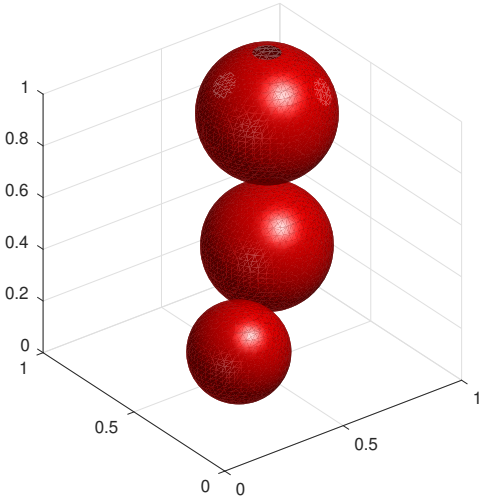


Fig. 3.1: An instance of 3D random potential field using Gaussian wells. The isosurface is at level -0.5.

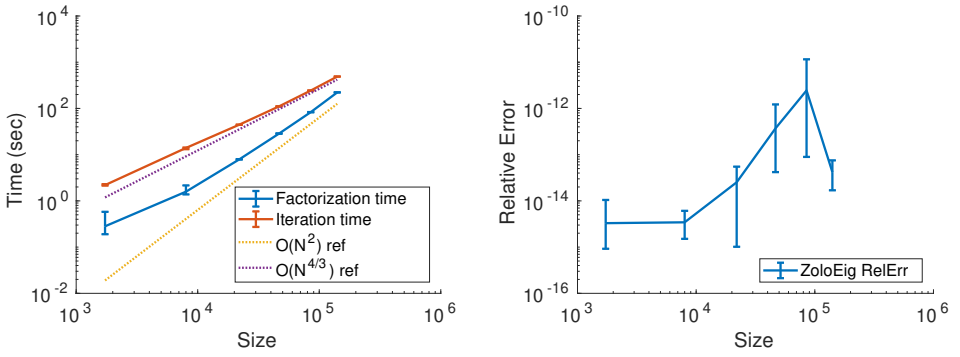


Fig. 3.2: The running time and the relative error for 3D Hamiltonian operator with Gaussian wells solved via ZoloEig. The relative error here is the relative error of eigenpairs defined in (3.1).

1,728 to 140,608. The order r is (4,4) in the ZoloEig. For each matrix, we provide the true eigenvalues $\lambda_1, \lambda_{88}, \lambda_{89}$ as the input, $a_- = -\infty, a_+ = \lambda_1, b_- = \lambda_{88}, b_+ = \lambda_{89}$, where λ_1 is the smallest eigenvalue, λ_{88} and λ_{89} are the 88th and 89th small eigenvalues. The ZoloEig is executed 5 times with different initial random vectors for each matrix. In Figure 3.2a and Figure 3.2b, these results are presented in a bar plot manner: the vertical bars indicate the largest and the smallest values, whereas the trend line goes through the mean values. And Table 3.2 shows the means of the results across 5 runs.

As we can read from Figure 3.2a, the iteration time for ZoloEig scales as $N^{4/3}$ while the factorization time scales as N^2. Both of them agree with the scaling of multifrontal method. Although for the examples here, the iteration time is more expensive than the factorization time, as N getting larger, the total runtime will quickly be dominated by the factorization. Therefore, reducing the number of factorizations would significantly reduce the cost of the algorithm. Figure 3.2b shows the relative error of the eigenvalues, which in general increases mildly as the problem size increases. All the relative errors are achieved with only one subspace iteration. At the same time, we find that the errors are far smaller than the tolerance 10^{-8}. This implies that setting $r=(4,4)$ overkills the problem and, in practice, user could use smaller r.

3.2. Hamiltonian of Silicon Bulk

The second example is a sparse Hermitian definite matrix pencil, (A,B), generated by SIESTA (a quantum chemistry software). For a silicon bulk in 3D with y^3 supercell of cubic Si, a DZP basis set with radius 4 Å is adopted to discretize the system, where $y=2,3,4,5$. In the spectrum slicing problem, the interval is chosen to contain the smallest 93 eigenvalues. The ZoloEig algorithm with $r=(3,3)$, $n_{ss}=94$ is used to solve the eigenvalue problem. The tolerance for both ZoloEig and FEAST is set to be 10^{-14}. Here we discuss the choice of the parameters used in FEAST, as in Table 3.4, in detail. Since we want to keep the number of factorizations as small as possible, we test FEAST with fixed $n_{ss}=200$ and gradually increasing r starting from 3 until the first r that FEAST converges. Later, given the r, we choose n_{ss} that minimize n_{solv}. Therefore, we have tried our best to obtain the optimal parameters for FEAST to maintain the smallest possible number of factorizations.

y	N	δ_λ	r	$e_{\Lambda,X}$	n_{ss}	n_{iter}	n_{gmres}	n_{solv}	T_{fact}	T_{iter}	T_{total}
2	832	9.6e-02	(3,3)	4.7e-17	94	1	10	2820	1.1e+00	2.4e+00	3.5e+00
3	2808	3.5e-01	(3,3)	9.9e-16	94	1	7	1974	9.4e+00	7.9e+00	1.7e+01
4	6656	3.3e-02	(3,3)	3.8e-15	94	1	11	3102	4.4e+01	4.2e+01	8.6e+01
5	13000	9.0e-02	(3,3)	5.9e-15	94	1	9	2538	1.6e+02	8.0e+01	2.4e+02

Table 3.3: Numerical results of ZoloEig for generalized eigenvalue problems from SIESTA. y is the number of unit cell on each dimension and other notations are as in Table 3.2.

	ZoloEig						FEAST				
y	r	$e_{\Lambda,X}$	n_{ss}	n_{iter}	n_{gmres}	n_{solv}	r	$e_{\Lambda,X}$	n_{ss}	n_{iter}	n_{solv}
2	(3,3)	4.7e-17	94	1	10	2820	3	8.3e-15	97	29	8439
3	(3,3)	9.9e-16	94	1	7	1974	4	3.4e-15	94	19	7144
4	(3,3)	3.8e-15	94	1	11	3102	6	8.2e-15	96	11	6336
5	(3,3)	5.9e-15	94	1	9	2538	7	9.4e-15	112	9	7056

Table 3.4: Comparison between ZoloEig and FEAST in the example of SIESTA. Notations are as in Table 3.2

Table 3.3 includes the detail information of the numerical results of ZoloEig. According to column T_{fact} and T_{iter}, we find the same scaling as in the first example. However, the factorization time is more expensive here due to the increase of the nonzeros in the Hamiltonian. And the total time is dominated by the factorization when

$N = 13000$. Therefore, it is worth to emphasize again that reducing the number of factorizations is important.

Table 3.4 provides the comparison between ZoloEig and FEAST in the sequential cases. Note that these two algorithms were implemented in different programming languages: ZoloEig is implemented in MATLAB and FEAST is in Fortran. Direct comparison of the runtime is unfair for ZoloEig, since MATLAB code is usually about 5x to 10x slower than Fortran code [3]. Hence, we compare the total number of linear system solves here, which is the main cost of both algorithms besides the factorizations. Comparing two columns of n_{solv}'s in Table 3.4, we see that ZoloEig is about 2 to 3 times cheaper than FEAST in terms of the number of applying the direct solver, n_{solv}. More importantly, when the problem size is large, the factorization time of the direct solver is dominanting the runtime. In this regime, ZoloEig might be also more efficient than FEAST since it requires a smaller number of factorizations. ZoloEig requires only $r = 3$ factorizations in Table 3.4, while FEAST requires 3 to 7 factorizations and the number of factorizations slightly increases as the problem size grows.

In the case of parallel computing, spectrum slicing algorithms including both FEAST and ZoloEig could be highly scalable. For example, eigenpairs in different spectrum ranges can be estimated independently; multishift linear systems can be solved independently; and each equation solver can be applied in parallel. If there was unlimited computer resource, then the advantage of ZoloEig over FEAST in terms of a smaller number of factorization might be less significant, but still meaningful because the number of iterations $n_{iter}n_{gmres}$ in ZoloEig (considering both the GMRES iterations and subspace iterations) is smaller than the number of subspace iterations n_{iter} in FEAST, and these iteration numbers cannot be reduced by parallel computing. Therefore, if unlimited computer resource was used, the total parallel runtime will be dominated by the iteration time in both ZoloEig and FEAST, and hence ZoloEig could be still faster than FEAST. Note that in practice the computer resource might be limited. In such a case, it is of interest to design faster parallel algorithms with a fixed number of processes. Given a fixed number of processes, ZoloEig has less number of matrix factorization and hence can assign more processes to each matrix factorization and each application of the factorization. Therefore, the runtime of parallel matrix factorization and iterative part in ZoloEig would be shorter than that of FEAST. The parallel version of ZoloEig is under development and it is worth to expore this benifit for large-scale eigenvalue problems.

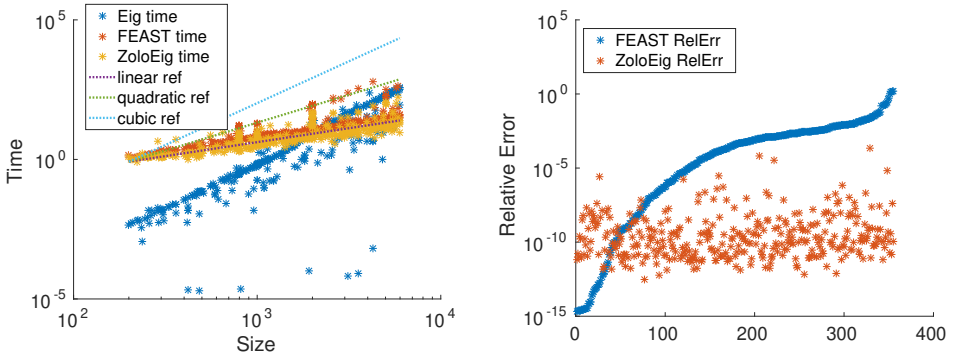
3.3. Florida Sparse Matrix Collection

In the third example, the proposed algorithm is applied to general sparse Hermitian matrices from the Florida sparse matrix collection. In order to show the broad applicability of the algorithm, all Hermitian matrices with size between 200 and 6,000 in the collection are tested. The full list of these matrices can be found in the test file "test_eigs_Florida.m" in the MATLAB toolbox. For each of these matrices, we randomly choose an interval (a,b) containing 96 eigenvalues.

In these examples, we compare the performance of the ZoloEig algorithm with the FEAST algorithm based on the contour integral method with trapezoidal rule. The subspace refinement is turned off again, aiming at testing the approximation accuracies of the Zolotarev's rational function and the discretized contour integral. The order r in the Zolotarev's rational function is 4 and the contour integral method has 16 poles.

[3] Even though there is difference between programming languages, we find that the actual runtime of ZoloEig is still faster than that of FEAST for large problem sizes, namely when $y \geq 4$ in Table 3.4.

Hence, both the ZoloEig and FEAST algorithms use the same order of rational functions in the approximation.



(a) (b)

Fig. 3.3: Running time and relative error for matrices in Florida matrix collection solved via ZoloEig and FEAST. The relative error here is the relative error of eigenvalues defined in (3.1).

Figure 3.3 visualizes the results of both the ZoloEig and the FEAST algorithms. Figure 3.3a includes the total running time of the MATLAB default dense eigensolver Eig, FEAST and ZoloEig. The running time of Eig aligns with the cubic scaling reference, whereas the running times of both FEAST and ZoloEig align with the linear scaling reference. As explained in previous examples, for a problem of small size, the iterative part in both FEAST and ZoloEig dominates the running time. The outliers of each line in these figures are caused by different sparsity densities and patterns of sparse matrices. According to Figure 3.3a, the running time of FEAST is constantly larger than ZoloEig. In Figure 3.3b, the relative error of FEAST is larger than ZoloEig for most matrices. Based on the right part of Figure 3.3b, FEAST fails for some sparse matrices, where the relative error is close to 1. Meanwhile, the relative error of ZoloEig is smaller than 1e-4 in all cases and the overall accuracy is about 1e-10. This observation supports that the composition of Zolotarev's rational functions is a better way to approximate rectangular functions.

4. Conclusion

This paper proposed an efficient method for computing selected eigenpairs of a sparse Hermitian definite matrix pencil (A,B) in the generalized eigenvalue problem. First, based on the best rational function approximations of signum functions by Zolotarev, the best high-order rational filter in a form of function compositions is proposed. Second, taking advantage of the shift-invariant property of Krylov subspaces in iterative methods and the matrix sparsity in sparse direct solvers, a hybrid fast algorithm is proposed to apply the best rational filter in the form of function compositions. Assuming that the sparse Hermitian matrices A and B are of size $N\times N$ and contains $O(N)$ nonzero entries, the computational cost for computing $O(1)$ eigenpairs is $O(F_N)$, where F_N is the operation count for solving the shifted linear system $(A-\sigma B)x=b$ using sparse direct solvers.

Comparing to the state-of-the-art algorithm FEAST, the proposed ZoloEig has a

better performance in our test examples for sequential computation. The numerical results in the sequential computation also implies that ZoloEig might also have good performance in parallel computation, which will be left as future work.

It is worth pointing out that the proposed rational filter can also be applied efficiently if an efficient dense direct solver or an effective iterative solver for solving the multi-shift linear systems in (2.35) is available. The proposed rational function approximation can also be applied as a preconditioner for indefinite sparse linear system solvers [32] and the orbital minimization method in electronic structure calculation [14]. These will be left as future works.

Acknowledgments. Y. Li was supported in part by National Science Foundation under awards DMS-1454939 and OAC-1450280, and also AMS-Simons Travel Grant. H. Yang was partially supported by the US National Science Foundation under award DMS-1945029. The authors would like to thank Fabiano Corsetti for seting up Silicon Bulk examples.

Proofs of the properties in Section 2.3 **Proof of Proposition 2.1:**

Proof. First we prove that we have the following partial fraction representation

$$Mx\prod_{j=1}^{r-1}\frac{x^2+c_{2j}}{x^2+c_{2j-1}} = Mx\left(1+\sum_{j=1}^{r-1}\frac{b_j}{x^2+c_{2j-1}}\right), \tag{.1}$$

where

$$b_j = (c_{2j}-c_{2j-1})\prod_{k=1,k\neq j}^{r-1}\frac{c_{2k}-c_{2j-1}}{c_{2k-1}-c_{2j-1}} \tag{.2}$$

for $j=1,\ldots,r-1$. Since any rational function has a partial fraction form and the coefficients $\{c_{2j-1}\}$ are distinct, Equation (.1) holds. One can verify (.1) by multiplying x^2+c_{2j-1} to both sides and set $x=\imath\sqrt{c_{2j-1}}$.

By (.1), we have

$$Z_{2r}(x;\ell) = Mx\frac{\prod_{j=1}^{r-1}(x^2+c_{2j})}{\prod_{j=1}^{r}(x^2+c_{2j-1})} = Mx\left(1+\sum_{j=1}^{r-1}\frac{b_j}{x^2+c_{2j-1}}\right)\frac{1}{x^2+c_{2r-1}}. \tag{.3}$$

Hence, simple partial fraction representations of

$$\frac{b_j}{(x^2+c_{2j-1})(x^2+c_{2r-1})} = \frac{b_j}{c_{2r-1}-c_{2j-1}}\left(\frac{1}{x^2+c_{2j-1}}-\frac{1}{x^2+c_{2r-1}}\right) \tag{.4}$$

for $j=1,\ldots,r-1$ complete the proof of the proposition. ∎

Proof of Proposition 2.2:

Proof. We further decompose (2.27) as complex rational functions,

$$Z_{2r}(x;\ell) = M\sum_{j=1}^{r}\frac{a_j}{2}\left(\frac{1}{x+\imath\sqrt{c_{2j-1}}}+\frac{1}{x-\imath\sqrt{c_{2j-1}}}\right). \tag{.5}$$

Substitute the Möbius transformation into (.5),

$$\begin{aligned} Z_{2r}(T(x);\ell) &= M\sum_{j=1}^{r}\frac{a_j}{2}\left(\frac{x-\beta}{\gamma(x-\alpha)+\imath\sqrt{c_{2j-1}}(x-\beta)}+\frac{x-\beta}{\gamma(x-\alpha)-\imath\sqrt{c_{2j-1}}(x-\beta)}\right) \\ &= M\sum_{j=1}^{r}\frac{a_j}{2}\left(\frac{\frac{x-\beta}{\gamma+\imath\sqrt{c_{2j-1}}}}{x-\frac{\gamma\alpha+\imath\sqrt{c_{2j-1}}\beta}{\gamma+\imath\sqrt{c_{2j-1}}}}+\frac{\frac{x-\beta}{\gamma-\imath\sqrt{c_{2j-1}}}}{x-\frac{\gamma\alpha-\imath\sqrt{c_{2j-1}}\beta}{\gamma-\imath\sqrt{c_{2j-1}}}}\right). \end{aligned} \tag{.6}$$

We denote

$$\sigma_j := \frac{\gamma\alpha + \imath\sqrt{c_{2j-1}}\beta}{\gamma + \imath\sqrt{c_{2j-1}}} = \frac{(\gamma^2\alpha + c_{2j-1}\beta) + \imath\sqrt{c_{2j-1}}(\beta-\alpha)\gamma}{\gamma^2 + c_{2j-1}}. \tag{.7}$$

Readers can verify that

$$\bar{\sigma}_j = \frac{\gamma\alpha - \imath\sqrt{c_{2j-1}}\beta}{\gamma - \imath\sqrt{c_{2j-1}}}, \tag{.8}$$

where $\bar{\sigma}_j$ is the complex conjugate of σ_j. Equation (.6) can be rewritten as,

$$Z_{2r}(T(x);\ell) = M\sum_{j=1}^{r}\frac{a_j\gamma}{\gamma^2 + c_{2j-1}} + M\sum_{j=1}^{r}\left(\frac{w_j}{x-\sigma_j} + \frac{\bar{w}_j}{x-\bar{\sigma}_j}\right), \tag{.9}$$

where

$$w_j = \frac{a_j(\sigma_j - \beta)}{2(\gamma + \imath\sqrt{c_{2j-1}})}. \tag{.10}$$

□

REFERENCES

[1] N. I. Akhiezer. *Theory of approximation.* F. Unger Pub. Co., New York, 1956. 2.1, 2.2

[2] N. I. Akhiezer. *Elements of the theory of elliptic functions.* American Mathematical Soc., 1990. 2.2, 2.1

[3] H. M. Aktulga, L. Lin, C. Haine, E. G. Ng, and C. Yang. Parallel eigenvalue calculation based on multiple shiftinvert Lanczos and contour integral based spectral projection method. *Parallel Comput.*, 40(7):195–212, 2014. 1, 1.1

[4] D. Braess. On rational approximation of the exponential and the square root function. In *Ration. Approx. Interpolat.*, pages 89–99. Springer Berlin Heidelberg, 1984. 1.2

[5] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Trans. Math. Softw.*, 9(3):302–325, sep 1983. 1.2, 2.3

[6] A. A. Gončar. Zolotarev Problems Connected with Rational Functions. *Math. USSR-Sbornik*, 7(4):623–635, apr 1969. 2.2, 2.2

[7] S. Güttel, E. Polizzi, P. T. P. Tang, and G. Viaud. Zolotarev Quadrature Rules and Load Balancing for the FEAST Eigensolver. *SIAM J. Sci. Comput.*, 37(4):A2100–A2122, jan 2015. 1.1, 1.2, 2.2, 3

[8] N. Hale, N. J. Higham, and L. N. Trefethen. Computing A^α, $\log(A)$, and Related Matrix Functions by Contour Integrals. *SIAM J. Numer. Anal.*, 46(5):2505–2523, jan 2008. 1.1

[9] K. L. Ho and L. Ying. Hierarchical interpolative factorization for elliptic operators: differential equations. *Commun. Pure Appl. Math.*, 2015. 2.3

[10] R. Li, Y. Xi, E. Vecharynski, C. Yang, and Y. Saad. A Thick-Restart Lanczos Algorithm with Polynomial Filtering for Hermitian Eigenvalue Problems. *SIAM J. Sci. Comput.*, 38(4):A2512–A2534, jan 2016. 1, 1.1

[11] Y. Li and L. Ying. Distributed-memory Hierarchical Interpolative Factorization. *Preprint*, 2016. 2.3

[12] L. Lin, J. Lu, L. Ying, and W. E. Pole-based approximation of the Fermi-Dirac function. *Chinese Ann. Math. Ser. B*, 30:729–742, nov 2009. 1.1

[13] J. W. H. Liu. The multifrontal method for sparse matrix solution: theory and practice. *SIAM Rev.*, 34(1):82–109, mar 1992. 1.2, 2.3

[14] J. Lu and H. Yang. Preconditioning orbital minimization method for planewave discretization. *SIAM Multiscale Modeling and Simulation*, to appear. 4

[15] D. A. Mazziotti. Towards idempotent reduced density matrices via particle-hole duality: McWeeny's purification and beyond. *Phys. Rev. E*, 68(6):066701, dec 2003. 1.2

[16] Y. Nakatsukasa, Z. Bai, and F. Gygi. Optimizing Halley's Iteration for Computing the Matrix Polar Decomposition. *SIAM J. Matrix Anal. Appl.*, 31(5):2700–2720, jan 2010. 1.2

[17] Y. Nakatsukasa and R. W. Freund. Computing Fundamental Matrix Decompositions Accurately via the Matrix Sign Function in Two Iterations: The Power of Zolotarev's Functions. *SIAM Rev.*, 58(3):461–493, jan 2016. 1.2, 2.2

[18] A. M. N. Niklasson. Expansion algorithm for the density matrix. *Phys. Rev. B*, 66(15):155115, oct 2002. 1.2

[19] I. Ninomiya. Best rational starting approximations and improved Newton iteration for the square root. *Math. Comput.*, 24:391–404, 1970. 1.2

[20] A. H. R. Palser and D. E. Manolopoulos. Canonical purification of the density matrix in electronic-structure theory. *Phys. Rev. B*, 58(19):12704–12711, nov 1998. 1.2

[21] P. P. Petrushev and V. A. Popov. *Rational approximation of real functions*. Cambridge University Press, Cambridge, 1987. 2.2, 2.2

[22] E. Polizzi. Density-matrix-based algorithm for solving eigenvalue problems. *Phys. Rev. B*, 79(11):115112, mar 2009. 1, 1.1, 1.1, 2.1, 2.2, 3

[23] Y. Saad. *Iterative methods for sparse linear systems*, volume 8 of *Stud. Comput. Math.* Society for Industrial and Applied Mathematics, second edition, 2003. 2.3

[24] T. Sakurai and H. Sugiura. A projection method for generalized eigenvalue problems using numerical integration. *J. Comput. Appl. Math.*, 159(1):119–128, 2003. 1, 1.1

[25] T. Sakurai and H. Tadano. CIRR: a Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems. *Hokkaido Math. J.*, 36(4):745–757, nov 2007. 1, 1.1

[26] P. G. Schmitz and L. Ying. A fast nested dissection solver for Cartesian 3D elliptic problems using hierarchical matrices. *J. Comput. Phys.*, 258:227–245, 2014. 2.3

[27] G. Schofield, J. R. Chelikowsky, and Y. Saad. A spectrum slicing method for the Kohn-Sham problem. *Comput. Phys. Commun.*, 183(3):497–505, 2012. 1, 1.1

[28] P. T. P. Tang, J. Kestyn, and E. Polizzi. A new highly parallel non-Hermitian eigensolver. In *Proc. High Perform. Comput. Symp.*, pages 1–9. Society for Computer Simulation International, 2014. 1.1, 2.1, 2.2

[29] M. Van Barel. Designing rational filter functions for solving eigenvalue problems by contour integration. *Linear Algebra Appl.*, 502:346–365, 2016. 1, 1.1, 1.1

[30] M. Van Barel and P. Kravanja. Nonlinear eigenvalue problems and contour integrals. *J. Comput. Appl. Math.*, 292:526–540, 2016. 1.1

[31] Y. Xi and Y. Saad. Computing Partial Spectra with Least-Squares Rational Filters. *SIAM J. Sci. Comput.*, 38(5):A3020–A3045, jan 2016. 1, 1.1, 1.1

[32] Y. Xi and Y. Saad. A rational function preconditioner for indefinite sparse linear systems. *SIAM Journal on Scientific Computing*, to appear. 4

[33] J. Xia. Efficient structured multifrontal factorization for general large sparse matrices. *SIAM J. Sci. Comput.*, 35(2):A832–A860, 2013. 2.3

[34] X. Ye, J. Xia, R. H. Chan, S. Cauley, and V. Balakrishnan. A fast contour-integral eigensolver for non-Hermitian matrices. Technical report, 2016. 1, 1.1, 1.1, 2.1, 2.2

[35] E. Zolotarev. Application of elliptic functions to questions of functions deviating least and most from zero. *Zap. Imp. Akad. Nauk. St. Petersbg.*, 30(5):1–59, 1877. 2.2