A RATIONAL FUNCTION PRECONDITIONER FOR INDEFINITE SPARSE LINEAR SYSTEMS *

YUANZHE XI† AND YOUSEF SAAD†

Abstract. This paper introduces a rational function preconditioner for linear systems with indefinite sparse matrices A. By resorting to rational functions of A, the algorithm decomposes the spectrum of A into two disjoint regions and approximates the restriction of A^{-1} on these regions separately. We show a systematic way to construct these rational functions so that they can be applied stably and inexpensively. An attractive feature of the proposed approach is that the construction and application of the preconditioner can exploit two levels of parallelism. Moreover, the proposed preconditioner can be modified at a negligible cost into a preconditioner for a near-by matrix of the form $A - cI$, which can be useful in some applications. The efficiency and robustness of the proposed preconditioner are demonstrated on a few tests with challenging model problems, including problems arising from the Helmholtz equation in three dimensions.

Key words. Rational function, incomplete LU, deflation, Cauchy integral, approximate inverse, Helmholtz equation

AMS subject classifications. 15A06, 65F08, 65F10, 65N22

1. Introduction.

In this paper, we consider iterative methods for solving large sparse linear systems

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$. To solve such systems, Krylov subspace methods preconditioned with a form of incomplete LU (ILU) factorization are often advocated because they generally achieve a good compromise between efficiency and robustness. However, classical ILU factorizations become less reliable for indefinite matrices. For example, these methods often fail for the discretized Helmholtz equation with high frequency in seismic imaging simulations [23], or for shifted linear systems encountered in the computation of interior eigenvalue problems [53]. They also generally fail for saddle point matrices such as those obtained from mixed finite element methods in fluid and solid mechanics [6]. Matrices that arise from such applications are often highly indefinite and ILU factorizations will either encounter some small/zero pivots during the factorization process or produce unstable triangular factors [15, 17]. Many attempts have been made in the past to overcome these difficulties. Among these we cite the work on Multilevel ILUs [5, 12, 40, 51, 55], on inverse based ILUs [8, 9, 11], on analytic ILUs [24, 25] and the more recent techniques that combine ILUs with low-rank corrections [35, 63]. The performance of these methods is still often not satisfactory for highly indefinite matrices.

The modified ILU factorization (MILU) [30, 43] was originally developed as an improved version of ILU that consisted of lumping all, or only a fraction, of the elements dropped during the elimination process and adding them to the diagonal entries of the computed factors [52]. Later, a diagonal modification idea was developed that consisted of perturbing the diagonal entries of the original matrix prior to performing the factorization in the numerical solution of Helmholtz problems [10, 21, 39, 41, 47, 60]. For indefinite matrices, these modified forms of ILU, which perturb the diagonal entries by complex numbers, tend to

*This work was supported by NSF under grants DMS-1216366 and DMS-1521573 and by the Minnesota Supercomputing Institute

†Address: Department of Computer Science & Engineering, University of Minnesota, Twin Cities. {yxi,saad}@cs.umn.edu

yield much more stable factors [1] than those obtained by simply increasing the accuracy (allowing more fill-ins) of ILU on the original matrix. Larger perturbations will yield more stable factors but these factors will likely be poor approximations to the original matrix A resulting in much slower convergence.

The preconditioner presented in this paper can be viewed as a generalization of the diagonal perturbation technique just mentioned, which attempts to approximate A^{-1} by exploiting rational functions. This preconditioner exhibits improved robustness relative to existing ILU-type preconditioners. In addition, both the construction and application procedures can exploit the multilevel parallelism provided by modern computing architectures. The main contributions of the paper are as follows.

1. General indefinite sparse matrices. Theoretical results regarding diagonal modification strategies are often limited to Hermitian matrices and they impose many constraints on the choice of perturbations. These requirements are dropped in this paper by resorting to a rational function approximation framework [32, 33]. In a nutshell, if P is the eigenprojector associated with the eigenvalues of A inside a circle Γ in the complex plane, we decompose A^{-1} into the sum of $(I-P)A^{-1}$ and PA^{-1} and approximate these two terms separately. We derive the Cauchy integral representation of $(I-P)A^{-1}$ and approximate it by numerical integration into a linear combination of shifted inverses $(A-\sigma_i I)^{-1}$. The contour Γ and the quadrature rules are selected in a systematic way so that the resulting poles σ_i improve the diagonal dominance property of A in order to yield an inexpensive and accurate ILU factorization for each $A-\sigma_i I$. To approximate PA^{-1} or its multiplication with a vector, we propose two schemes for different scenarios. If the number of eigenvalues inside Γ is not very large, we compute these eigenpairs by a FEAST-like [48] scheme and approximate PA^{-1} with the computed spectral information. Otherwise, an inner-outer iteration scheme is presented to efficiently approximate the matrix-vector product associated with PA^{-1}. In both cases the ILU factors computed in the rational approximation of $(I-P)A^{-1}$ are reused. The derivation of the preconditioner does not rely on any form of symmetry of the underlying system. However, if A is Hermitian, the cost of constructing and applying the preconditioner can be reduced by half due to the complex conjugate property of the poles.

2. Multi-level parallelism. The preconditioner proposed in this paper can exploit two levels of parallelism in both the construction and application phases, making it attractive for modern high performance computing architectures. Parallelism of the operations across different poles σ_i constitutes the first level of parallelism. The other level corresponds to the use of domain decomposition techniques to parallelize the ILU factorization and subsequent triangular solves [36] associated with each shifted matrix $A-\sigma_i I$.

3. Updating the preconditioner. In many applications, a sequence of linear systems whose coefficient matrices differ by a moderate real diagonal shift need to be solved. One typical example is Helmholtz equations discretized on the same mesh but with respect to different frequencies in seismic imaging simulations. To obtain a preconditioner for a shifted matrix $A-cI$, where c is a small shift, it is not easy to update an ILU factorization that has been previously computed for A. Often, the only alternative is to recompute a new ILU factorization for each new matrix and this is costly. In contrast, the cost of updating the proposed rational function preconditioners for these situations is negligible (it costs $O(1)$ operations), and this will work provided c is not too large.

The rest of the paper is organized as follows. Section 2 introduces the basic idea of the rational function preconditioner. A few practical issues are discussed in Section 3 and

[1] Here we adopt a common abuse of language: if the condition number of the computed ILU factors is relatively small, we say that these factors are 'stable', otherwise, they are 'unstable'. This terminology was first introduced by H.C. Elman in [17].

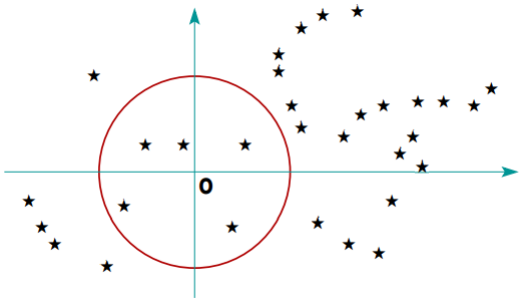

FIG. 2.1. *Standard approach to rational function approximations: Cauchy integral approach. Stars represent eigenvalues and the circle encloses the desired part of the spectrum.*

numerical results are presented in Section 4. Finally, concluding remarks are drawn in Section 5.

2. Rational approximation of A^{-1}. This section describes how to build a rational function preconditioner that combines ILU factorizations and a form of deflation, which is based on (2.14).

2.1. Background: Cauchy integral representation of eigenprojectors. The use of rational functions has been tied to the Cauchy integral representation of the eigenprojector to compute eigenpairs of matrices. This is represented in the work by Sakurai and co-workers [56, 57] and by Polizzi in the context of the FEAST package [48]. Given a circle Γ enclosing a desired part of the spectrum, the eigenprojector associated with the eigenvalues inside the circle is given by

$$P = \frac{1}{2i\pi}\int_{\Gamma}(sI - A)^{-1}ds, \tag{2.1}$$

where the integration is performed counter-clockwise [53]. See Figure 2.1 for illustration.

If a numerical integration scheme is used, then P will be approximated by a certain linear operator $\tilde{P}$ that takes the form

$$\tilde{P} = \frac{1}{2}\sum_{k=1}^{2m}\tilde{\alpha}_k(A - \tilde{\sigma}_k I)^{-1}. \tag{2.2}$$

This rational function of A is then used instead of A in a Krylov subspace or subspace iteration algorithm as a filter to compute desired eigenpairs. A few details on this approach will be provided in Section 2.3.

2.2. Rational approximation of $(I - P)A^{-1}$. Let P be the eigenprojector defined in (2.1). Then one can in theory compute $Pf(A)$ for a function f via the formula:

$$Pf(A) = \frac{1}{2i\pi}\int_{\Gamma}(sI - A)^{-1}f(s)ds, \tag{2.3}$$

where the integration is performed counter-clockwise. The condition here is that f has to be analytic in the region enclosed by Γ. We cannot therefore apply this idea to the function $f(z) = 1/z$ when Γ encloses the origin which is the situation of interest for preconditioning.

However, since $1/z$ is analytic outside Γ, it is feasible to calculate A^{-1} restricted to the spectrum *outside* Γ. Without loss of generality, we assume Γ is a circle with center at the

origin and radius r to simplify the analysis in this section. More general contour shapes can be analyzed in a similar way and will be discussed in Section 3. By making the change of variables $t = 1/s$ in (2.3), we get

$$(I-P)A^{-1} = \frac{1}{2i\pi}\int_{\Gamma'_-}\left(\frac{1}{t}I - A\right)^{-1} \times t \times \frac{-dt}{t^2} = \frac{1}{2i\pi}\int_{\Gamma'}(I-tA)^{-1}dt, \tag{2.4}$$

where Γ'_- (resp. Γ') denotes the circle with center at the origin and radius $1/r$ running clock-wise (resp. counter-clockwise). Applying a numerical integration scheme, we end up with the approximation:

$$(I-P)A^{-1} \approx \frac{1}{2}\sum_{k=1}^{2p}\alpha_k(I-\sigma_k A)^{-1}. \tag{2.5}$$

Because $(I-P)$ is singular, this cannot be used as a preconditioner in theory. However, recall that the above is only an approximation and the right-hand side of (2.5) is unlikely to be singular. Therefore, we could potentially use it as a preconditioner but our aim is different.

A few details about the weights α_k and poles σ_k used in (2.5) are now discussed. Substituting A and I in the right-hand side of (2.4) with a complex variable z and the integer 1, respectively, we obtain the corresponding scalar function

$$h(z) = \frac{1}{2i\pi}\int_{\Gamma'}\frac{1}{(1-tz)}dt. \tag{2.6}$$

Based on the Cauchy integral formula, we know that

$$h(z) = \begin{cases} 0 & |z| < r \\ 1/z & |z| > r \end{cases}. \tag{2.7}$$

Note also that $h(z)$ takes real values when z is real.

We now exploit the change of variables $t = e^{i\pi x}/r$ to get:

$$h(z) = \frac{1}{2}\int_{-1}^{1}\frac{e^{i\pi x}/r}{(1-ze^{i\pi x}/r)}dx = \frac{1}{2}\int_{0}^{1}\frac{e^{i\pi x}/r}{(1-ze^{i\pi x}/r)}dx + \frac{1}{2}\int_{0}^{1}\frac{e^{-i\pi x}/r}{(1-ze^{-i\pi x}/r)}dx.$$

Any quadrature formula

$$\int_0^1 g(x)dx \approx \sum_{k=1}^{p}\omega_k g(x_k)$$

can be used and this will lead to

$$h(z) \approx \frac{1}{2}\sum_{k=1}^{p}\omega_k\frac{e^{i\pi x_k}/r}{(1-ze^{i\pi x_k}/r)} + \frac{1}{2}\sum_{k=1}^{p}\omega_k\frac{e^{-i\pi x_k}/r}{(1-ze^{-i\pi x_k}/r)} \equiv \frac{1}{2}\sum_{k=1}^{2p}\frac{\alpha_k}{1-z\sigma_k}, \tag{2.8}$$

where we have set for convenience

$$\alpha_k = \begin{cases} \omega_k e^{i\pi x_k}/r & k = 1,\ldots,p \\ \omega_{k-p}e^{-i\pi x_{k-p}}/r & k = p+1,\ldots,2p \end{cases}, \tag{2.9}$$

and

$$\sigma_k = \begin{cases} e^{i\pi x_k}/r & k = 1, \ldots, p \\ e^{-i\pi x_{k-p}}/r & k = p+1, \ldots, 2p \end{cases} . \tag{2.10}$$

It is easy to see that if no pole is along the real axis, the first p poles would be located in the upper half circle and related to the remaining ones by

$$\sigma_k = \overline{\sigma_{p+k}}, \quad k = 1, \ldots, p.$$

A similar relation also holds for the weights α_k. As a result, if A is real symmetric or complex Hermitian, the calculation of (2.5) can be simplified by using only the poles with positive (or negative) imaginary parts:

$$\frac{1}{2}\sum_{k=1}^{2p} \alpha_k (I - \sigma_k A)^{-1} = \Re e \sum_{k=1}^{p} \alpha_k (I - \sigma_k A)^{-1} = \Re e \sum_{k=p+1}^{2p} \alpha_k (I - \sigma_k A)^{-1}. \tag{2.11}$$

Applying the preconditioner defined by (2.5) (or (2.11)) requires solving linear systems with the matrix $I - \sigma_k A$. In order to reduce the resulting computational cost, we compute the ILU factorization

$$(\sigma_k^{-1} I - A) \approx L_k U_k, \tag{2.12}$$

and further approximate $(I - P)A^{-1}$ as

$$(I - P)A^{-1} \approx \frac{1}{2}\sum_{k=1}^{2p} \frac{\alpha_k}{\sigma_k} U_k^{-1} L_k^{-1}. \tag{2.13}$$

The approximation error of the right-hand side of (2.13) is dominated by two factors: (i) the quadrature rule used and (ii) the accuracy of the computed ILU factorizations.

Considering the second factor, it is known that the accuracy and numerical stability of ILU factorizations depend on the diagonal dominance of the matrix [15, 50, 52]. Thus, for a more diagonally dominant matrix, an ILU factorization with the same threshold may drop more fill-ins and still produce an effective preconditioner. Diagonal dominance of matrices of the form $A - \sigma_k^{-1} I$ is highly affected by the location of the poles σ_k^{-1}. In particular, it was shown in [47] that adding a purely imaginary part to a given real shift, will usually yield a better ILU factorization. As the shift σ_k^{-1} moves farther away from the real axis, the quality of the preconditioner starts to deteriorate since the ILU factorization of $A - \sigma_k^{-1} I$ approximates a matrix that is no longer close to A. One of the goals of the paper [47] is to devise heuristic compromises to optimize the selection of the shift. Note that the idea of using complex shifts has been exploited in many different ways before they were adapted to ILU factorizations, see, for example [21, 22, 27, 42].

With this in mind, we compare the location of the poles obtained from using the Gauss-Legendre rule, the Gauss-Chebyshev rule of the first kind, and the standard mid-point rule in Figure 2.2. We set r in (2.10) to 1 so that all the poles are on the unit circle. Detailed formulas for (α_k, x_k) associated with these rules can be found in Appendix A. Figure 2.2 shows that the poles for the mid-point rule have larger imaginary parts than those of the Gauss rules. The mid-point rule does not do a particularly good job at approximating integrals when compared to a Gaussian quadrature rule, but it can lead to more stable ILU factorizations and this is crucial to the overall performance of the preconditioner. The other observation that can be made is that the poles tend to concentrate near the real axis as the

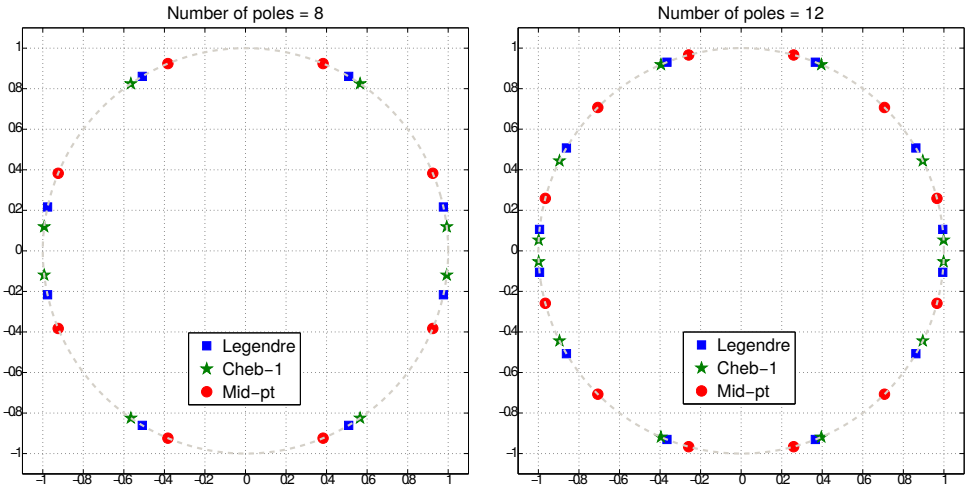

FIG. 2.2. *The location of the poles for the Gauss-Legendre rule, the Gauss-Chebyshev rule of the first kind, and the mid-point rule using* $2p$ *poles to approximate* $h(z)$ *in (2.6) with the radius* r *being set to* 1. *Left:* $p = 4$, *right:* $p = 6$.

number of poles increases. This suggests that it may be preferable to avoid using a large number of poles in this case.

Consider now the approximation error related to numerical quadrature by the mid-point rule. We plot the resulting approximations of $h(z)$ in Figure 2.3 by varying the number of poles used. The range of the variable z is inside the interval $[0, 8]$ so that $h(z)$ takes real values. Figure 2.3 indicates that using a relatively small number of poles, e.g., 4 poles on

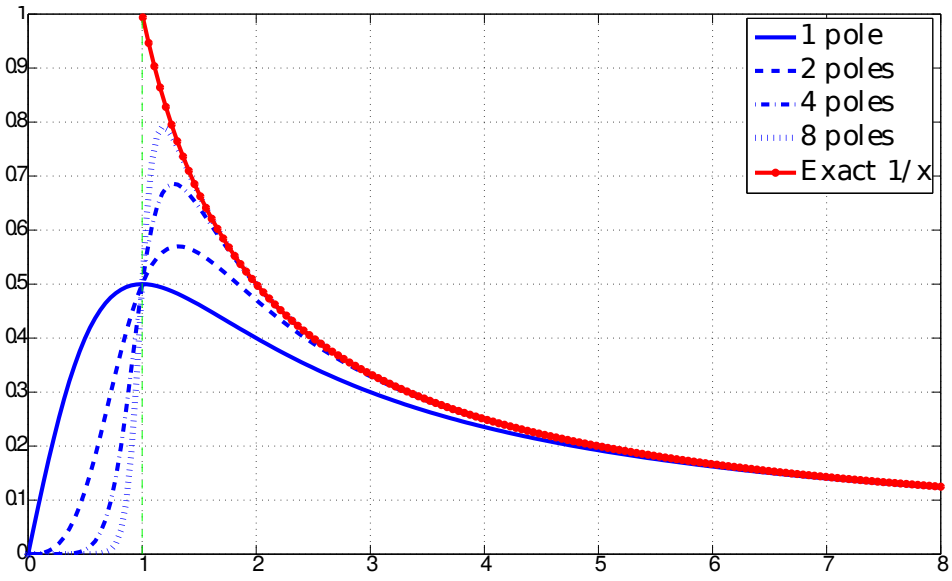

FIG. 2.3. *Comparison of the numerical approximations of* $h(z)$ *in (2.6) and* $1/z$ *on the interval* $[1, 8]$. *The radius* r *is set to* 1 *and the mid-point rule is used with* p *poles on the upper half plane, where* $p = 1, 2, 4, 8$.

the upper half plane, can still lead to a reasonable approximation of $1/z$ outside the circle Γ. Thus, the mid-point rule may achieve a good balance between a stable ILU factorization

and an accurate approximation of $h(z)$. As a result, we will often adopt the 8-pole mid-point rule in the construction of the preconditioner.

2.3. Approximation of PA^{-1}. In practice a preconditioner based on an approximation of $(I-P)A^{-1}$ will perform poorly. Indeed, $(I-P)A^{-1}$ approximates a singular operator even though its approximation obtained from quadrature may be nonsingular. A second issue is that the preconditioner aims at resolving eigen-components related to eigenvalues outside the circle Γ and leaving those inside Γ untouched. However, there may be a large number of eigenvalues inside Γ and the related components must also be reduced. This section explores remedies for this problem.

From (2.13) we have

$$A^{-1} = (I-P)A^{-1} + PA^{-1} \approx \frac{1}{2}\sum_{k=1}^{2p} \frac{\alpha_k}{\sigma_k} U_k^{-1} L_k^{-1} + PA^{-1}. \tag{2.14}$$

If we have an orthogonal basis Q of the invariant subspace associated with the enclosed eigenvalues, we can approximate PA^{-1} by $QC^{-1}Q^H$ where $C = Q^HAQ$ and obtain an approximate inverse of A. The invariant subspace can be readily computed by a FEAST-like procedure [34, 48].

The basic idea of the FEAST algorithm [48] is to apply subspace iteration to an approximate eigenprojector $\tilde{P}$ as defined in (2.2) to accelerate the convergence of the eigenpairs enclosed by Γ. In the original FEAST derivation [48, 59], $\tilde{P}$ was obtained via a quadrature approximation of an indicator function $p(z)$ represented by:

$$p(z) = \frac{1}{2i\pi}\int_\Gamma \frac{1}{s-z} ds.$$

Based on the Cauchy integral formula, we know that

$$p(z) = \begin{cases} 1 & |z| < r \\ 0 & |z| > r \end{cases}.$$

If a quadrature rule with $2m$ poles is used, we would get

$$p(z) \approx \frac{1}{2}\sum_{k=1}^{2m} \frac{\tilde{\alpha}_k}{z-\tilde{\sigma}_k}, \tag{2.15}$$

where $\tilde{\alpha}_k$ and $\tilde{\sigma}_k$ are defined in a similar way to (2.9)–(2.10) with r and p replaced by $1/r$ and m, respectively. Accordingly, $\tilde{P}$ will take the following form

$$\tilde{P} = \frac{1}{2}\sum_{k=1}^{2m} \tilde{\alpha}_k (A - \tilde{\sigma}_k I)^{-1}. \tag{2.16}$$

This $\tilde{P}$ will map all the eigenvalues of A enclosed by Γ close to 1 and the rest close to 0.

There has been several recent improvements made to the original FEAST algorithm. For example, for Hermitian problems, the Zolotarev rational function approximation approach was developed in [31] to yield the sharpest possible decrease from the plateau of one inside the target interval to zero outside. This leads to a faster convergence rate for the subspace iteration algorithm on which FEAST is based but one should note that the filter is mostly geared towards situations when direct methods are used to solve the shifted linear systems. Indeed, it was observed that these poles tended to concentrate near the real axis, rendering

iterative solution techniques slow or ineffective. In another example, the paper [64] argued that how well the step function is approximated is unimportant for the convergence and advocated a least-squares (LS) approximation approach to obtain rational functions that achieve a good balance between the quality of the approximation and the efficiency of the linear system solution by iterative methods. Several rational functions are plotted in Figure 2.4 as an illustration.

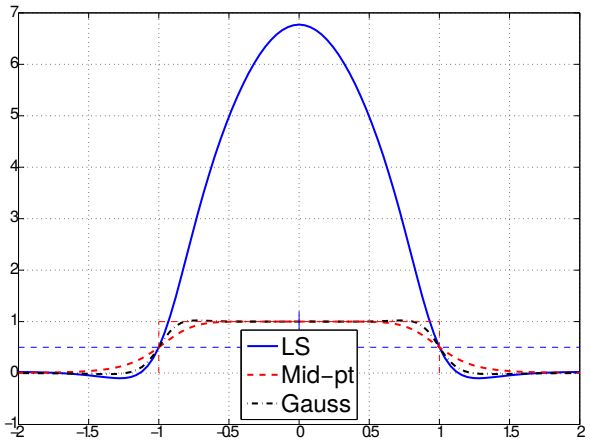

FIG. 2.4. *Comparison of a standard rational function based on the Cauchy integral using the mid-point rule with 4 poles on the upper half plane (dashed line), a least-squares (LS) rational function (solid line) using the same poles, and a standard Cauchy integral using Gaussian quadrature with 4 poles (dash-dotted line) on the reference interval* $[-1, 1]$.

Figure 2.4 indicates that the Cauchy-Gauss based rational function (dash-dotted line) yields a better approximation to the indicator function $p(z)$ than the mid-point based rational function (dashed line) when both of them use 4 poles on the upper half plane. Although the LS rational function (solid line) based on the same poles as the mid-point rational function does not approximate 1 as well as the other two inside $[-1, 1]$, it yields the sharpest decrease at the boundaries, -1 and 1, and thus is likely to lead to the fastest convergence for subspace iteration [64]. This suggests that a good strategy to build the filter, is to combine the $\tilde{\sigma}_k$ from the mid-point rule and the $\tilde{\alpha}_k$ from the LS approach.

Once poles and weights are specified, the FEAST algorithm amounts to applying subspace iteration in conjunction with a Rayleigh-Ritz procedure to $\tilde{P}$. This is summarized in Algorithm 1.

The most expensive operation in Algorithm 1 is the solution of $2m$ linear systems in Lines 4–6. Next, we will show that it is possible to reuse the ILU factorizations computed in (2.12) to solve these linear systems if $\tilde{\sigma}_k$ is chosen in an appropriate way.

Taking the reciprocal of σ_k in (2.10) leads to

$$\sigma_k^{-1} = \begin{cases} re^{-i\pi x_k} & k = 1, \ldots, p \\ re^{i\pi x_{k-p}} & k = p+1, \ldots, 2p \end{cases} . \tag{2.17}$$

This shows that if the same number of poles and the same quadrature rule are applied in (2.2) and (2.5), the following relation will hold:

$$\tilde{\sigma}_k = \begin{cases} \sigma_{k+p}^{-1} & k = 1, \ldots, p \\ \sigma_{k-p}^{-1} & k = p+1, \ldots, 2p \end{cases} . \tag{2.18}$$

1: Input: a block $Q \in \mathbb{C}^{n\times s}$ of initial eigenvector approximations
2: $Its := 0$, $U := [\ ]$,
3: **while** $Its \leq MaxIts$ **do**
4: **for** $k = 1, \ldots, 2m$ **do**
5: Compute $U := U + \tilde{\alpha}_k(\tilde{\sigma}_k I - A)^{-1}Q$
6: **end for**
7: Orthonormalize the columns of U
8: Compute $\Theta = U^H AU$
9: Compute the Schur decomposition $\Theta = VCV^H$
10: Update $Q = UV$
11: Set $Its := Its + 1$
12: **end while**

ALGORITHM 1
A FEAST-like Filtered Subspace Iteration.

Therefore, we obtain

$$(\tilde{\sigma}_k I - A) \approx \begin{cases} L_{k+p}U_{k+p} & k = 1, \ldots, p \\ L_{k-p}U_{k-p} & k = p+1, \ldots, 2p \end{cases}. \tag{2.19}$$

As a consequence, we can reuse the ILU factorizations computed for $(\sigma_k^{-1} I - A)$ in a number of ways. For example, if the ILU factors have been computed with enough accuracy, the linear systems in Steps 4-6 in Algorithm 1 can be directly solved with a forward and a backward substitution. Otherwise, the factors can serve as preconditioners for an iterative scheme for solving the same systems. Moreover, similar to (2.11), if A is Hermitian, Steps 4-6 in Algorithm 1 can be simplified by performing only the first m steps of the for loop.

3. Practical issues. This section discusses a few techniques to improve the performance of the preconditioner proposed in the previous section.

3.1. Shifting the center. We first consider an artifice to improve diagonal dominance of each shifted matrix by moving the center of Γ away from the origin. Note that the Cauchy integral representation (2.4) of $(I - P)A^{-1}$, and its approximation (2.5), are still valid as long as Γ encloses the origin.

We may assume without loss of generality that most of the eigenvalues of A lie on the right side of the complex plane. We would like to move the circle to the left so that it contains fewer eigenvalues with positive real parts. If the matrix is Hermitian, $(I-P)A^{-1}$ would have no negative eigenvalues when the circle is large enough and this operation would be somewhat equivalent to solving the linear system without the presence of negative eigenvalues. As a rule, we will move the center of the circle horizontally to the left *in such a way that the rightmost quadrature pole will lie on the imaginary axis.* For example, since the 8-pole midpoint rule shown in Figure (2.2) has its rightmost poles located at $(0.9239, \pm 0.3827)$, the whole circle is then shifted to the left by 0.9239. Figure 3.1 illustrates this.

We now need to derive the Cauchy integral representation of $(I - P)A^{-1}$ for a circle that is centered at $c \neq 0$, i.e., for $\Gamma = \{z : |z - c| = r\ \}$ with $|c| < r$. Making the change of variables $t = \frac{1}{z-c}$, any point z outside Γ corresponds to a point t with $|t| < 1/r$. Therefore:

$$\begin{aligned}(I - P)A^{-1} &= \frac{1}{2i\pi}\int_{\Gamma'_-} \left(\left(\frac{1}{t} + c\right) I - A\right)^{-1} \times \frac{t}{1+tc} \times \frac{-dt}{t^2} \\ &= \frac{1}{2i\pi}\int_{\Gamma'} \left(\left(\frac{1}{t} + c\right) I - A\right)^{-1} \times \frac{1}{t + t^2 c} dt,\end{aligned}$$

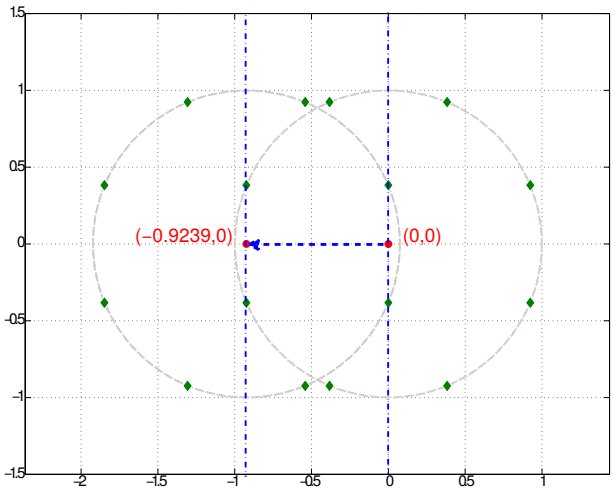

FIG. 3.1. *Shifting the center of* Γ *in Figure 2.2 from* $(0,0)$ *to* $(-0.9239,0)$ *such that all the quadrature poles (diamonds) for the mid-point rule have non-positive real parts.*

where $\Gamma_-^{'}$ (resp. $\Gamma^{'}$) denotes the circle with center at the origin and radius $1/r$ running clock-wise (resp. counter-clockwise). The change of variables $t = e^{i\pi x}/r$ results in:

$$(I-P)A^{-1} = \frac{1}{2}\int_{-1}^{1}((re^{-i\pi x}+c)I-A)^{-1} \times \frac{1}{e^{i\pi x}/r + ce^{2i\pi x}/r^2} e^{i\pi x}/r dx.$$

When approximating the above integral with a standard quadrature rule, we obtain the following:

$$(I-P)A^{-1} \approx \frac{1}{2}\sum_{k=1}^{2p} \frac{\alpha_k}{\sigma_k + c\sigma_k^2} \times ((\sigma_k^{-1}+c)I-A)^{-1}. \tag{3.1}$$

Accordingly, the approximation of the eigenprojector P associated with the eigenvalues inside Γ is now given by:

$$P \approx \frac{1}{2}\sum_{k=1}^{2p} \tilde{\alpha}_k (A-(\tilde{\sigma}_k+c)I)^{-1}. \tag{3.2}$$

Here, α_k, σ_k, $\tilde{\alpha}_k$ and $\tilde{\sigma}_k$ are defined in the same way as in (2.9), (2.10) and (2.15), respectively. Moreover, it is easy to see that the following relation holds:

$$A-(\tilde{\sigma}_k+c)I = \begin{cases} A-(\sigma_{k+p}^{-1}+c)I & k=1,\ldots,p \\ A-(\sigma_{k-p}^{-1}+c)I & k=p+1,\ldots,2p \end{cases}. \tag{3.3}$$

This means that the approximations of $(I-P)A^{-1}$ in (3.1) and P in (3.2) can still share the ILU factorizations as was the case when $c=(0,0)$. The construction of the rational function preconditioner is summarized in Algorithm 2.

We note that the loop starting in Line 4 of the algorithm can be performed in parallel. The ILU factorization of each $A-(\sigma_k^{-1}+c)I$ is computed in Line 5. The algorithm then calculates two scalars a_k and b_k in Lines 6-7, which correspond to the coefficients in front of $(A-(\sigma_k^{-1}+c)I)^{-1}$ in (3.1) and (3.2), respectively. If A is Hermitian, half of the ILU factorizations are needed and the k loop will stop after p iterations.

1: Input: radius r of Γ, number of quadrature poles $2p$
2: Output: ILU factors $\{L_k, U_k\}$, two vectors a, b
3: Compute α_k, σ_k via (2.9)-(2.10) and center c
4: **for** $k = 1, \ldots, 2p$ **do**
5: Compute $A - (\sigma_k^{-1} + c)I \approx L_k U_k$ via ILU factorization
6: Compute $a_k = -\alpha_k/(2\sigma_k + 2c\sigma_k^2)$
7: Compute $b_k = \overline{\alpha_k} r^2/2$
8: **end for**

ALGORITHM 2
Construction of the rational function preconditioner M^{-1}.

3.2. Avoiding forming the basis of the invariant subspace explicitly. One issue with the procedure described in Section 2.3 is that we may have many eigenvalues inside Γ and deflating them would require computing and storing a large orthonormal basis of the related invariant subspace. Since the application of the preconditioner in (2.14) essentially only requires the evaluation of matrix-vector products of the form $PA^{-1}v$, the second improvement to our proposed scheme is to directly approximate $PA^{-1}v$ rather than PA^{-1}.

The first observation is that P and A^{-1} commute, i.e., $PA^{-1} = A^{-1}P$. This implies $PA^{-1}v = A^{-1}Pv$ in theory, i.e., when the projector P is exact. If we denote $A^{-1}Pv$ by z, then an approximation of z can be computed by applying a residual-minimizing Krylov subspace method to solve the problem

$$\min_z \|Pv - Az\|_2. \tag{3.4}$$

In an actual implementation, P in (3.4) is replaced by its approximation $\tilde{P}$ in (3.2) and GMRES [54] is called to solve for z. Since the exact solution z is in the range of P, we seek an approximate solution of the form $z = \tilde{P}y$, i.e., we solve

$$A\tilde{P}y = \tilde{P}v, \quad z = \tilde{P}y. \tag{3.5}$$

This essentially amounts to using $\tilde{P}$ as a right preconditioner to solve $Az = \tilde{P}v$ by GMRES. The advantage of this approach is that the spectrum of $A\tilde{P}$ is more clustered than that of A, and this is favorable to a Krylov subspace method such as GMRES.

The application of the rational function preconditioner M^{-1} to a vector v is summarized in Algorithm 3. Note that the loop starting in line 3, can be trivially parallelized.

1: Input: $z^{(1)} := 0$, $z^{(2)} := 0$, $g := 0$, v, a, b and $\{L_k, U_k\}$
2: Output: z
3: **for** $k = 1, \ldots, 2p$ **do**
4: Compute $g \approx (A - (\sigma_k^{-1} + c)I)^{-1}v$
5: Update $z^{(1)} := z^{(1)} + a_k g$
6: Update $z^{(2)} := z^{(2)} + b_k g$
7: **end for**
8: Apply m steps of GMRES without restart to solve $Az^{(3)} = z^{(2)}$ via (3.5)
9: Compute $z = z^{(1)} + z^{(3)}$

ALGORITHM 3
Application of the rational function preconditioner M^{-1} to a vector v.

We would like to comment on Line 4 of the above algorithm. As discussed at the end of Section 3.1, L_k, U_k computed in Line 5 of Algorithm 2 can be used in two different ways to

solve $(A-(\sigma_k^{-1}+c)I)x = v$. When the accuracy of the ILU factorization is high, $U_k^{-1}(L_k^{-1}v)$ will immediately yield a good approximate solution. Otherwise, L_k and U_k can be used as preconditioners for GMRES to solve this linear system. The same comment can be made for the case when applying $\tilde{P}$ as the right preconditioner in Line 8. We will compare these two approaches in Section 4 with a few numerical examples.

3.3. Exploiting information from inner iterations. Since the application of the preconditioner involves the use of GMRES in Step 8 of Algorithm 3, this rational function preconditioner belongs to the class of flexible inner-outer Krylov methods [3, 29, 46, 49, 58, 61]. An appealing feature of these methods is that some approximate eigenvectors can be estimated from inner iterations and then used to accelerate the convergence of outer GMRES iterations.

Suppose the FGMRES framework [49] is exploited in both the inner and outer iterations. Then the outer iteration will generate one basis $V_m \equiv \{v_1, v_2, \ldots, v_m\}$ of the Krylov subspace $\mathcal{K}_m(A, r_0)$ with $r_0 = b - Ax_0$ and the other basis $Z_m \equiv \{z_1, z_2, \ldots, z_m\}$ of the solution subspace, which satisfy the following relation:

$$AZ_m = V_{m+1}\bar{H}_m,$$

where $\bar{H}_m$ is upper Hessenberg of size $(m+1) \times m$. An approximate solution x_m is of the form

$$x_m = x_0 + Z_m y_m,$$

where y_m is the minimizer of

$$\min_{y\in\mathbb{C}^m} \|b - AZ_m y\|_2 = \min_{y\in\mathbb{C}^m} \|V_{m+1}(\beta e_1 - \bar{H}_m y)\|_2 = \min_{y\in\mathbb{C}^m} \|\beta e_1 - \bar{H}_m y\|_2,$$

and e_1 is the first canonical basis vector. At each outer iteration i, FGMRES is invoked to compute $z_i = M^{-1}v_i$, which corresponds to Line 8 of Algorithm 3 with $z^{(2)}$ instantiated as $\tilde{P}v_i$:

$$Az^{(3)} = \tilde{P}v_i. \tag{3.6}$$

One way to speed up the convergence of subsequent inner FGMRES is to resort to deflation. This technique has been well studied in the literature [4, 13, 16, 20, 28, 44, 45]. In this paper, we follow the method proposed by Morgan [44] to augment the Krylov subspace with approximate eigenvectors $u_1, u_2, \ldots, u_p$, so that the solution x_m belongs to

$$x_0 + \text{span}\{r_0, A\tilde{P}r_0, \ldots, (A\tilde{P})^{m-p-1}r_0, u_1, u_2, \ldots, u_p\}.$$

The modification of Morgan's algorithm to allow for right preconditioning is given in [13]. For the sake of completeness, we summarize it in Algorithm 4.

Note that eigenvectors are computed in Step 12 of Algorithm 4. In the first outer iteration, no eigenvectors are available and p is set to $p = 0$. In subsequent iterations, we can compute the p smallest eigenpairs[2] (θ_i, g_i) of the following generalized eigenvalue problem [13]

$$\bar{H}_m^T\bar{H}_m g = \theta\bar{H}_m^T V_{m+1}^T W_m g,$$

[2]Here we adopt a common abuse of language: the smallest eigenpairs are the eigenpairs associated with the smallest eigenvalues.

1: Compute $r_0 = b$, $\beta = ||r_0||_2$ and $v_1 = r_0/\beta$
2: **for** $j = 1, \ldots, m$ **do**
3: Set $w_j = \begin{cases} v_j & k = 1, \ldots, m-p \\ u_{j-m+p} & k = m-p+1, \ldots, m \end{cases}$
4: Compute $z := A\tilde{P}w_j$
5: For $i = 1, \ldots, j$, do $\begin{cases} h_{i,j} = (z, v_i) \\ z := z - h_{i,j}v_i \end{cases}$
6: Compute $h_{j+1,j} = ||z||_2$ and $v_{j+1} = z/h_{j+1,j}$
7: **end for**
8: Define $V_{m+1} := [v_1, \ldots, v_{m+1}]$, $W_m = [w_1, \ldots, w_m]$
9: Compute $x_m = x_0 + \tilde{P}W_m y_m$, where $y_m = \arg\min_y ||\beta e_1 - \bar{H}_m y||_2$
10: If satisfied stop
11: $x_0 \leftarrow x_m$
12: Compute p eigenvectors $u_1, \ldots, u_p$ of $A\tilde{P}$, and go to 1

ALGORITHM 4
Right preconditioned FGMRES with deflation.

and set $u_i = W_m g_i$. These u_is are called Harmonic Ritz vectors. As an alternative, one can compute the p eigenpairs (θ_i, y_i) of largest magnitude of the problem:

$$H_m y = \theta V_m^T W_m y,$$

and set $u_i = W_m y_i$, where H_m is the Hessenberg matrix obtained from $\bar{H}_m$ by removing its last row. These u_is are called Ritz vectors. Comparisons between these two schemes are provided in Section 4.

3.4. Solving slightly shifted systems. Another appealing property of the proposed rational function preconditioner is that it is possible to modify a preconditioner that has been computed for A, into a preconditioner for $A - c_1 I$ where c_1 satisfies certain conditions. Furthermore, this modification entails essentially no additional computational cost.

The main idea behind this technique is based on the fact that the circle $\Gamma(c, r)$ with center c and radius r can actually be shifted 'by a different amount', to $\Gamma(c - c_1, r)$ for the new problem. As can be readily observed, the factorizations for A using the circle $\Gamma(c, r)$ and those for $A - c_1 I$ using the circle $\Gamma(c - c_1, r)$ are identical. This is because

$$(A - c_1 I) - (\sigma_k^{-1} + \ (c - c_1))I = A - (\sigma_k^{-1} + c)I.$$

Looking at Algorithm 2, we immediately see that if we apply it to the matrix $A - c_1 I$ with a circle $\Gamma(c - c_1, r)$, then indeed the L_k, U_k factors are identical as observed above, and that the only change is in computing the scalars a_k in Line 6. The cost for computing these new values of a_k is negligible. The only condition required to be able to use this technique is that the new circle should still enclose the origin. If c_1 is real this means that $|c - c_1|$ should be less than r.

The above discussion can be easily generalized to the case when there are s linear systems with coefficient matrices $A - c_i I$, where the real shifts c_i are ordered in descending order:

$$c_1 < c_2 < \cdots < c_s.$$

One can start by constructing a preconditioner for a reference starting matrix $A - c_t I$ with $1 \leq t \leq s$ using a circle $\Gamma(c, r)$ and then update the preconditioners for the remaining $s - 1$

linear systems as outlined above. Indeed, for a given matrix $A - c_j I$ we write $A - c_j I = (A - c_t I) - (c_j - c_t)I$ and so c_1 in the above discussion is replaced by $c_j - c_t$. Note that depending on the location of c_j relative to c_t, the circle will move to the right or the left and that the approach is valid for the j-th system as long as $|c - (c_j - c_t)| < r$.

4. Numerical examples. This section presents some numerical results to illustrate the performance of the rational function preconditioner. All algorithms were implemented in MATLAB and the tests were performed on a Linux cluster with 252 GB of memory. The ILU factorization was computed by the MATLAB bulit-in function ILUTP with a fixed threshold of 10^{-3}. The following notation is used throughout this section:

- fill: ratio of the number of non-zeros in the preconditioner over the number of non-zeros in the original matrix;
- p-t: wall clock time to build the preconditioner in seconds;
- its: number of iterations to reduce the initial residual by a desired factor;
- i-t: wall clock time in the iteration phase.

4.1. 3D shifted Laplacian. We begin our tests with the following model problem

$$\begin{aligned} -\Delta u - \eta u &= f \ \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{4.1}$$

where the PDEs are defined over $\Omega = (0,1)^3$ with the zero Dirichlet boundary conditions. When these PDEs are discretized by the 7-point stencil, the eigenvalues and eigenvectors of the resulting matrices can be determined explicitly [52].

4.1.1. Effect of the radius of Γ on preconditioning. We first study the effect of the radius of Γ on preconditioning. The test problem was obtained by discretizing (4.1) with $\eta = 640$ on a 40^3 grid. The resulting matrix has 232 negative eigenvalues. The matrix entries were scaled by $1/40^2$ and reordered by the approximate minimum degree (AMD) ordering [1, 2] to reduce the fill-ins during the factorization. The number of poles used in the rational function preconditioner was fixed at 8 and FGMRES(40) without restart was applied in the inner solves. Notice that since the test matrix is real symmetric, only those 4 poles in either half plane are needed in the actual computation. The outer FGMRES(40) iteration stopped when the relative residual norm was reduced by a factor of 10^5.

We then varied the radius of Γ and compared the iteration counts and the corresponding fill factors to solve the above linear system. Based on the discussion in the previous sections, we know that the preconditioning effect depends on two factors: 1) the accuracy of the approximation $\tilde{P}$ to the exact eigenprojector P; 2) the accuracy in the numerical solution of linear systems associated with $A\tilde{P}$ in the inner iterations. Let us first assume $\tilde{P}$ is an accurate approximation and study the second factor. This can be achieved by computing exact L,U factors in the factorizations to avoid inaccuracy from ILUs. In general, a larger radius will enclose more eigenvalues inside Γ and therefore this will make $A\tilde{P}$ harder to solve by GMRES. This is confirmed by the data in the second and third columns of Table 4.1. As can be seen, when r increases from 0.25 to 64, more eigenvalues fall inside Γ and, as expected, the iteration count increases from 2 to 13. This implies that a smaller radius leads to faster convergence of GMRES iterations when $\tilde{P}$ is a reasonable approximation to P.

Next we replaced the exact L,U factors with an ILUTP factorization and reran the experiments. The vector g in Line 4 of Algorithm 3 was computed by $U_k^{-1}(L_k^{-1}v)$. The corresponding iteration counts are tabulated in the fourth column of Table 4.1. It is observed that the iteration count does not increase monotonically as in the exact LU case. It first

decreases from 62 to 7 when the radius increases from 0.25 to 16 and then increases to 13 as the radius reaches 64. One possible explanation for this comes from the less stable ILU factorizations of the shifted matrices. Once the quadrature rule is fixed, the approximation accuracy of $\tilde{P}$ only depends on the accuracy of the ILU approximation to each shifted matrix. Since the test matrix is highly indefinite, a smaller radius will yield less stable ILUs, resulting in a larger error in the approximation to P. We also find that when the radius becomes larger than 16, the iteration counts remain the same whether an ILU or the exact LU are used. Indeed, in this case the ILU factorizations are almost as accurate as the exact LU due to the larger radii used.

Finally, the last column of Table 4.1 shows that the fill factor from the rational function preconditioner decreases monotonically from 39.66 to 5.82 as the radius increases. Compared with the fill factor of 92.93 from an exact LU factorization of the test matrix, the rational function preconditioner requires much less memory to solve the problem. In summary, the results in Table 4.1 suggest that a relatively large radius is preferable for a better performance of the rational function preconditioner when L_k, U_k are used to compute g directly. For example, the use of a radius of 16 results in the second smallest iteration count with the fill factor 9.78 in Table 4.1. When solving large indefinite linear systems, a heuristic rule of thumb is to select the radius by aiming to reach a fill factor of around 10.

radius	# of eigs. inside Γ	LU	ILU	
		its	its	fill
0.25	241	2	62	39.66
1	302	2	10	23.35
2	386	3	9	17.83
4	585	4	9	13.80
8	1063	6	7	11.84
16	2400	7	8	9.78
32	6547	11	11	6.64
64	25230	13	13	5.82

TABLE 4.1

Effects of the radius of Γ on preconditioning the 40^3 shifted Laplacian matrix. This test matrix has 232 negative eigenvalues. Both exact LU and ILUTP with threshold 10^{-3} are applied to compute L_k, U_k in Line 5 of Algorithm 2 and g in Line 4 of Algorithm 3 is computed by $U_k^{-1}(L_k^{-1}v)$. The outer FGMRES stops when the relative residual norm is reduced by a factor of 10^5.

Next we computed g in Line 4 of Algorithm 3 by another approach. We utilized a larger threshold of 0.05 in the ILU factorization in Step 5 of Algorithm 2 and tested three radii: 1, 2 and 4. The computed factors were combined with GMRES(5) and GMRES(10) without restart to solve the linear systems in Lines 4 and 8 of Algorithm 2. As can be seen from Table 4.2, the fill factors are reduced to around 5.4 and the outer iteration counts are much smaller compared with the cases in Table 4.1 when ILU factors are used to solve shifted linear systems directly.

As a comparison, we also applied the standard ILUTP preconditioner to solve the same test problem. Figure 4.1 shows the convergence profiles for standard ILUTP preconditioned GMRES(40) with three different thresholds (0.004, 0.005 and 0.006). As shown in the figure, even when the fill factor reaches 20.09, the ILUTP-preconditioned GMRES(40) still fails to converge to the desired tolerance. In particular, the three preconditioned methods all stagnated after 50 iterations and the residuals were reduced by a factor of less than 10^2 even after 400 iterations. The condition numbers of the L and U factors, as estimated by MATLAB'S `condest` function, were $\mathrm{cond}(L) = 1.00 \times 10^{10}$ and $\mathrm{cond}(U) = 2.68 \times 10^5$ for

radius	fill	GMRES(5)	GMRES(10)
1	5.40	11	3
2	5.38	6	3
4	5.37	5	4

TABLE 4.2
Effects of using L_k, U_k as preconditioners for GMRES(m) to solve linear systems associated with shifted linear systems in Lines 4 and 8 of Algorithm 3. The ILUTP threshold is set to 0.05 and the outer FGMRES stops when the relative residual norm is reduced by a factor of 10^5.

the case when the threshold is 0.004.

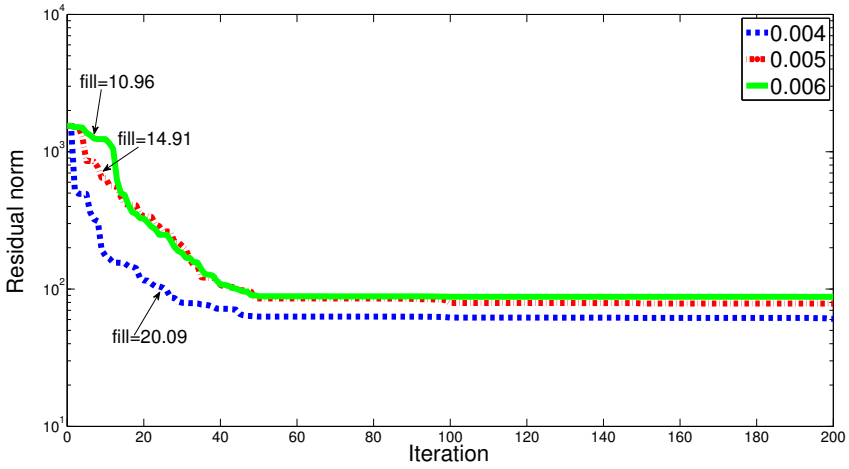

FIG. 4.1. *Convergence of standard ILUTP-preconditioned GMRES(40) with different thresholds (0.004, 0.005 and 0.006) on the 40^3 shifted Laplacian test problem in Section 5.1.1.*

Additional tests were performed for this problem with the *shifted* ILU preconditioner. Three preconditioners were constructed by applying ILUTP with the threshold 0.004 to factorize $A + sI$ for $s = 0.2i, 0.5i$ and $1.0i$. As shown in Figure 4.2, shifted ILUTP-preconditioned GMRES(40) only converges to the desired accuracy for $s = 0.2i$ in the first 200 iterations. The convergence profiles shown in this figure illustrate the well-known drawback of this approach, which has to do with the selection of s. A large s improves the numerical stability of the factorization and at the same time it leads to less fill-ins in the computed factors. However, the resulting factorization is a less effective preconditioner because it approximates a matrix that is far away from the original matrix. Another issue with this approach is the loss of self-adjointness since the matrix $A - sI$ is not Hermitian even when A is.

4.1.2. Effects of inner solves and deflation. We next fixed the radius in the construction of the preconditioner to 16 and varied the dimension of the inner FGMRES algorithm to solve the same test problem as the one of the previous section.

Table 4.3 shows that the outer iteration count decreases from 43 to 5 as the dimension increases. However, a larger Krylov subspace dimension takes more time for each outer iteration and the least iteration time is achieved when FGMRES(30) is used as the inner solver.

Next we tested the two deflation schemes discussed in Section 3.3 on the same problem. At the kth outer iteration, we injected $k-1$ Harmonic or Ritz vectors in the Krylov subspace

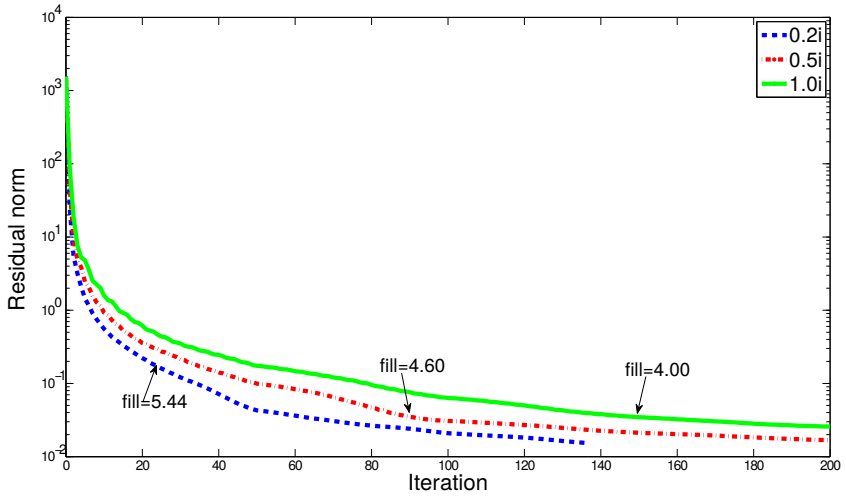

FIG. 4.2. *Convergence of modified ILUTP-preconditioned GMRES(40) with different diagonal shifts* ($0.2i$, $0.5i$ *and* $1.0i$) *on the* 40^3 *shifted Laplacian test problem in Section 5.1.1. The threshold used in ILUTP factorizations is* 0.004.

m	10	15	20	25	30	35	40	45	50	55	60
its	43	30	16	13	9	8	8	7	7	6	5
i-t	15.02	15.56	11.43	11.89	10.14	10.93	12.89	13.11	14.68	14.50	13.55

TABLE 4.3
Iteration counts and time when FGMRES(m) is used in inner solves.

generated by the inner FGMRES(m) algorithm. In order to avoid increasing the Krylov subspace dimension, we kept the dimension of the augmented subspace constant and equal to m. It was found that a better performance was obtained when Ritz vectors were used, in contrast to cases reported in [13, 44].

m	Harmonic		Ritz	
	its	i-t	its	i-t
40	7	12.39	7	12.29
50	6	14.03	5	11.85
60	6	17.09	5	14.30

TABLE 4.4
Effects of using deflation in the inner iteration. The number of vectors injected into the Krylov subspace at the kth inner solve is $k-1$ *and the dimension of the augmented subspace is fixed at* m.

4.2. 3D Helmholtz problem. The second test problem is the Helmholtz equation of the following form:

$$\left(-\Delta - \frac{\omega^2}{c(x)^2}\right) u(x,\omega) = s(x,\omega), \tag{4.2}$$

where Δ is the Laplacian, ω is the angular frequency, $c(x)$ is the seismic velocity field, and $u(x,\omega)$ is the time-harmonic wavefield solution to the forcing term $s(x,\omega)$. The domain of interest is the unit cube $D = (0,1)^3$. We assume that the mean of $c(x)$ is equal to 1. The

zero Dirichlet boundary condition was applied to one side and the Perfectly Matched Layer (PML) boundary condition [7, 14] was applied to the rest. We kept 8 points per wavelength when discretizing (4.2) by the 7-point stencil on $N \times N \times N$ grids. Due to the PML boundary condition, the discretized matrices were complex non-Hermitian. The forcing term s was generated by a Gaussian point source centered at $(1/2, 1/2, 1/2)$. The resulting discretized linear system and right hand side were scaled by h^2, where h is the grid space.

Since 3-digit accuracy is often sufficient in geophysics applications [18, 19, 37, 38, 62], we stopped the outer GMRES when the relative residual norm was reduced by a factor of 10^3. We used GMRES(35) without restart in the inner iteration and chose the radius of the circle Γ to be $r = 30$. The ILU factors were directly used to solve linear systems in Lines 4 and 8 of Algorithm 3. The number of poles used was fixed at 8. The ILU factorization of the 8 shifted matrices were parallelized by calling the MATLAB function `parfor`. The application of the preconditioner was performed in a serial fashion since the overhead of the `parfor` function is very costly in this case.

To show the scalability of the preconditioner, we tested the preconditioner on four problems discretized on grids with $N = 2^5, 2^6, 2^7, 2^8$. The matrices were reordered by the nested dissection (ND) ordering [26]. The level of the ND tree started with 9 when $N = 2^5$ and was increased by 3 when N doubled each time. The computational results are reported in Table 4.5.

$n = N^3$	$\omega/(2\pi)$	lev	fill	p-t	i-t	its
32^3	4	9	8.84	0.55	2.67	2
64^3	8	12	10.89	4.63	22.04	2
128^3	16	15	11.25	43.27	209.41	2
256^3	32	18	11.67	428.77	2059.10	2

TABLE 4.5

Numerical experiments for solving (4.2) on $N \times N \times N$ grids. The matrices are reordered by ND ordering with lev levels in the ND tree. GMRES(35) without restart is used in inner solves. The radius of the circle Γ is chosen as 30 and the number of poles is fixed at 8.

We first observe that the fill factor grows slowly to 11.67 as the problem size increases. Moreover, the construction time scales as $O(n \log n)$. The iteration number remains constant in these tests and the iteration time seems to scale like $O(n \log n)$ as well. These scaling results can be visualized in Figure 4.3 where a comparison with the curve $n \log n$ is provided.

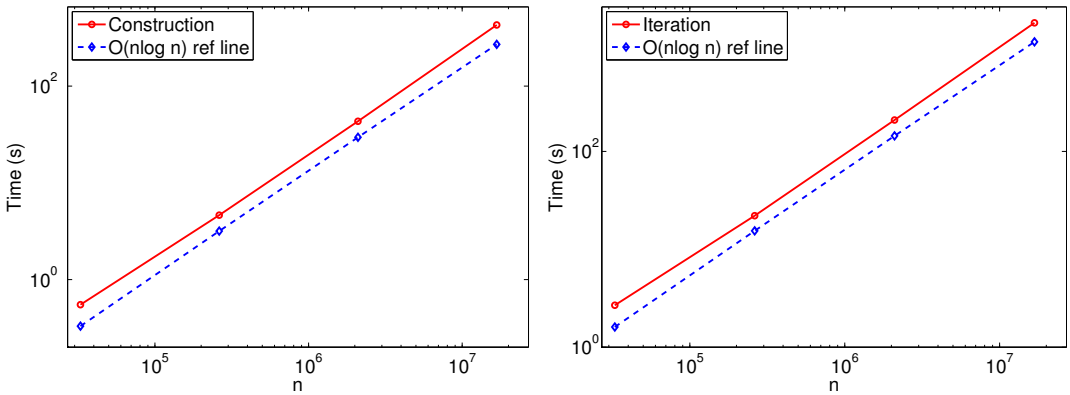

FIG. 4.3. *Time scaling results for the Helmholtz test problem in Table 4.5, comparing with the reference scaling of $n \log n$, where n is the test matrix size. Left: Construction time, right: Iteration time.*

TABLE 4.6
The fill factors of ILU factorizations of 8 shifted matrices as well as corresponding poles in the computation of the 128^3 Helmboltz problem in Table 4.5.

label	pole	$\text{nnz}(L_i + U_i)/\text{nnz(A)}$
1	$-00.00 + 11.48i$	2.18
2	$-16.24 + 27.72i$	1.14
3	$-39.20 + 27.71i$	1.14
4	$-55.43 + 11.48i$	1.14
5	$-00.00 - 11.48i$	2.21
6	$-16.24 - 27.72i$	1.14
7	$-39.20 - 27.71i$	1.14
8	$-55.43 - 11.48i$	1.14

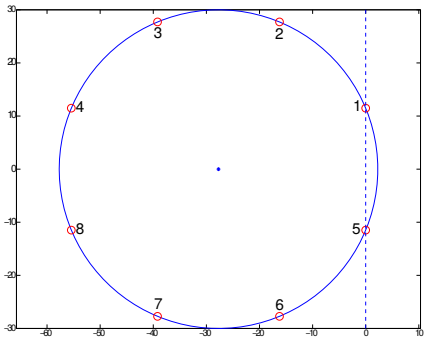

FIG. 4.4. *Illustration of the location of poles.*

We also reported the fill factor associated with each shifted matrix as well as the pole location in Table 4.6. It can be seen that the two shifted matrices with poles on the y-axis have denser factors than the remaining six matrices. Since the fill factor of each shifted matrices remains relatively small, the backward/forward triangular solves associated with them are as efficient as sparse matrix-vector products. The location of the poles are further illustrated in Figure 4.4.

5. Conclusion. This paper discussed a rational function preconditioner for indefinite sparse matrices, based on decomposing the inverse of the original matrix into two parts and approximating each part with rational functions of A that in turn exploit ILU factorizations. The numerical experiments show that for the same given fill factor, this preconditioner is far more robust than preconditioners based on ILU factorizations when solving linear systems with highly indefinite coefficient matrices. In an inner-outer scheme based on using a Krylov subspace method, it is observed that the number of outer GMRES steps achieved with this preconditioner is small and independent of the problem size and frequency for 3D Helmholtz equations discretized on regular grids. More work remains to be done to further improve the efficiency and robustness of the algorithm. For example, we plan on constructing a second level preconditioner to precondition $A\tilde{P}$ to reduce the inner iteration costs. We also plan to take full advantage of the two levels of parallelism of this preconditioner by implementing it on high performance computers. Finally, we would like to adapt this preconditioner to the solution of other types of indefinite problems such as those related to saddle point problems.

Acknowledgments. We would like to thank anonymous referees for their useful suggestions which led to substantial improvements of the original version of this paper. YX also would like to thank Xiao Liu for fruitful discussions about the Helmholtz tests.

REFERENCES

[1] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.
[2] ———, *Algorithm 837: An approximate minimum degree ordering algorithm*, ACM Trans. Math. Software, 30 (2004), pp. 381–388.
[3] O. AXELSSON AND P. S. VASSILEVSKI, *A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 625–644.
[4] J. BAGLAMA, D. CALVETTI, G. H. GOLUB, AND L. REICHEL, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1998), pp. 243–269.
[5] R. E. BANK AND C. WAGNER, *Multilevel ILU decomposition*, Numer. Math., 82 (1999), pp. 543–576.

[6] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.
[7] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Computat. Phys., 114 (1994), pp. 185 – 200.
[8] M. Bollhöfer, *A robust and efficient ILU that incorporates the growth of the inverse triangular factors*, SIAM J. Sci. Comput., 25 (2003), pp. 86–103.
[9] M. Bollhöfer, J. I. Aliaga, A. F. Martín, and E. S. Quintana-Ortí, *ILUPACK*, (2011), pp. 917–926.
[10] M. Bollhöfer, M. J. Grote, and O. Schenk, *Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media*, SIAM J. Sci. Comput., 31 (2009), pp. 3781–3805.
[11] M. Bollhöfer and Y. Saad, *Multilevel preconditioners constructed from inverse-based ILUs*, SIAM J. Sci. Comput., 27 (2005), pp. 1627–1650.
[12] E. F. F. Botta and F. W. Wubs, *Matrix renumbering ILU: An effective algebraic multilevel ILU preconditioner for sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1007–1026.
[13] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.
[14] W. C. Chew and W. H. Weedon, *A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates*, Microw. Opt. Techn. Let., 7 (1994), pp. 599–604.
[15] E. Chow and Y. Saad, *Experimental study of ILU preconditioners for indefinite matrices*, J. Comput. Appl. Math., 87 (1997), pp. 387–414.
[16] E. de Sturler, *Inner-outer methods with deflation for linear systems with multiple right-hand sides*, in In Householder Symposium XIII, Proceedings of the Householder Symposium on Numerical Algebra, Pontresina, Switzerland, June 17 - 26, 1996, pp. 193–196.
[17] H. C. Elman, *A stability analysis of incomplete LU factorizations*, Math. Comp., 47 (1986), pp. 191–217.
[18] B. Engquist and L. Ying, *Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation*, Commun. Pure Appl. Math., 64 (2011), pp. 697–735.
[19] B. Engquist and L. Ying, *Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers*, Multiscale Model. Simul., 9 (2011), pp. 686–710.
[20] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 303 – 318.
[21] Y.A. Erlangga, C. Vuik, and C.W. Oosterlee, *Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation*, Appl. Numer. Math., 56 (2006), pp. 648–666.
[22] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput., 27 (2005), pp. 1471–1492.
[23] O. G. Ernst and M. J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, vol. 83 of Lect. Notes Comput. Sci. Eng., Springer, Heidelberg, (2012), pp. 325–363.
[24] M. J. Gander and F. Nataf, *AILU: a preconditioner based on the analytic factorization of the elliptic operator*, Numer. Linear Algebra Appl., 7 (2000), pp. 505–526.
[25] M. J. Gander and F. Nataf, *AILU for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization*, J. Comput. Acoust., 9 (2001), pp. 1499–1506.
[26] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.
[27] M. B. Van Gijzen, Y. A. Erlangga, and C. Vuik, *Spectral analysis of the discrete helmholtz operator preconditioned with a shifted laplacian*, SIAM J. Sci. Comput., 29 (2007), pp. 1942–1958.
[28] L. Giraud, S. Gratton, X. Pinel, and X. Vasseur, *Flexible GMRES with deflated restarting*, SIAM J. Sci. Comput., 32 (2010), pp. 1858–1878.
[29] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput, 21 (1997), pp. 1305–1320.
[30] I. Gustafsson, *A class of first order factorization methods*, BIT, 18 (1978), pp. 142–156.
[31] S. Güttel, E. Polizzi, P. Tang, and G. Viaud, *Zolotarev quadrature rules and load balancing for the FEAST eigensolver*, SIAM J. Sci. Comput., 37 (2015), pp. A2100–A2122.
[32] N. Hale, N. J. Higham, and L. N. Trefethen, *Computing $A^\alpha, \log(A)$, and related matrix functions by contour integrals*, SIAM. J. Numer. Anal., 46 (2008), pp. 2505–2523.
[33] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, PA, USA, 2008.
[34] J. Kestyn, E. Polizzi, and P. Tang, *FEAST eigensolver for non-Hermitian problems*, arXiv:1506.04463 [math.NA], (2015).
[35] R. Li and Y. Saad, *Divide and conquer low-rank preconditioners for symmetric matrices*, SIAM J. Sci. Comput., 35 (2013), pp. A2069–A2095.
[36] Z. Li, Y. Saad, and M. Sosonkina, *pARMS: a parallel version of the algebraic recursive multilevel solver*, Numer. Linear. Algebra Appl., 10 (2003), pp. 485–509.

[37] F. Liu and L. Ying, *Additive sweeping preconditioner for the Helmholtz equation*, Multiscale Model. Simul., to appear, (2016).
[38] ———, *Recursive sweeping preconditioner for the 3D Helmholtz equation*, SIAM J. Sci. Comput., to appear, (2016).
[39] S. MacLachlan, D. Osei-Kuffuor, and Y. Saad, *Modification and compensation strategies for threshold-based incomplete factorizations*, SIAM J. Sci. Comput., 34 (2012), pp. A48–A75.
[40] S. MacLachlan and Y. Saad, *A greedy strategy for coarse-grid selection*, SIAM J. Sci. Comput., 29 (2007), pp. 1825–1853.
[41] M. Magolu monga Made, R. Beauwens, and G Warze, *Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix*, Comm. Numer. Methods Engrg., 16 (2000), pp. 801–817.
[42] M. Magolu Monga Made, R. Beauwens, and G. Warze, *Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix*, Comm. in Numer. Meth. in Engin., 16 (2000), pp. 801–817.
[43] T. A. Manteuffel, *Shifted incomplete Cholesky factorization*, in Sparse Matrix Proceedings 1978, SIAM, Philadelphia, (1979), p. 41?61.
[44] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[45] ———, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
[46] Y. Notay, *Flexible conjugate gradients*, SIAM J. Sci. Comput., 22 (2000), pp. 1444–1460.
[47] D. Osei-Kuffuor and Y. Saad, *Preconditioning Helmholtz linear systems*, Appl. Numer. Math., 60 (2010), pp. 420 – 431.
[48] E. Polizzi, *Density-matrix-based algorithm for solving eigenvalue problems*, Phys. Rev. B, 79 (2009), p. 115112.
[49] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[50] ———, *ILUT: A dual threshold incomplete LU factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.
[51] ———, *ILUM: A multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput., 17 (1996), pp. 830–847.
[52] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM Publications, Philadelphia, PA, 2nd ed., 2003.
[53] Y. Saad, *Numerical Methods for Large Eigenvalue Problems-revised edition*, SIAM, Philadelphia, 2011.
[54] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[55] Y. Saad and B. Suchomel, *ARMS: an algebraic recursive multilevel solver for general sparse linear systems*, Numer. Linear Algebra Appl., 9 (2002), pp. 359–378.
[56] T. Sakurai and H. Sugiura, *A projection method for generalized eigenvalue problems using numerical integration*, J. Comput. Appl. Math., 159 (2003), pp. 119 – 128. Japan-China Joint Seminar on Numerical Mathematics; In Search for the Frontier of Computational and Applied Mathematics toward the 21st Century.
[57] T. Sakurai and H. Tadano, *CIRR: a Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems*, Hokkaido Mathematical Journal, 36 (2007), pp. 745–757.
[58] V. Simoncini and D. B. Szyld, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2002), pp. 2219–2239.
[59] P. Tang and E. Polizzi, *FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 354–390.
[60] M.B. van Gijzen, Y.A. Erlangga, and C. Vuik, *Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian*, SIAM J. Sci. Comput., 29 (2007), pp. 1942–1958.
[61] C. Vuik, *New insights in GMRES-like methods with variable preconditioners*, J. Comput. Appl. Math., 61 (1995), pp. 189 – 204.
[62] S. Wang, M. V. de Hoop, and J. Xia, *On 3D modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver*, Geophys. Prospect., 59 (2011), pp. 857–873.
[63] Y. Xi, R. Li, and Y. Saad, *An algebraic multilevel low-rank preconditioner for sparse symmetric matrices*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 235–259.
[64] Y. Xi and Y. Saad, *Computing partial spectra with least-squares rational filters*, SIAM J. Sci. Comput., 38 (2016), pp. A3020–A3045.

Appendix A. Classical quadrature rules. In this section, we provide some details of several quadrature formulas to approximate the following integral

$$\int_0^1 g(x)dx \approx \sum_{k=1}^{p} \omega_k g(x_k).$$

The weights and poles for the mid-point rule have the form

$$\begin{cases} x_k &= \frac{(2k-1)}{2p} \\ w_k &= \frac{1}{p} \end{cases} \qquad k = 1, \cdots, p. \tag{A.1}$$

The Gauss-Chebyshev quadrature rule of the first kind uses the following weighs and poles:

$$\begin{cases} x_k &= \frac{1}{2}\left(1 + \cos\left(\frac{(2k-1)\pi}{2p}\right)\right) \\ w_k &= \frac{\pi}{2p}\sin\left(\frac{(2k-1)\pi}{2p}\right) \end{cases} \qquad k = 1, \cdots, p. \tag{A.2}$$

The poles and weights associated with the Gauss-Legendre rule are given by:

$$\begin{cases} x_k &= \frac{t_k+1}{2} \\ w_k &= \frac{1}{\left(1-t_k^2\right)[L_p'(t_k)]^2} \end{cases} \qquad k = 1, \cdots, p, \tag{A.3}$$

where t_k is the kth root of the p-th Legendre polynomial $L_p(x)$.