A CONTOUR-INTEGRAL BASED METHOD FOR COUNTING THE EIGENVALUES INSIDE A REGION IN THE COMPLEX PLANE

GUOJIAN YIN*

Abstract. In many applications, the information about the number of eigenvalues inside a given region is required. In this paper, we propose a contour-integral based method for this purpose. The new method is motivated by two findings. There exist methods for estimating the number of eigenvalues inside a region in the complex plane. But our method is able to compute the number of eigenvalues inside the given region exactly. An appealing feature of our method is that it can integrate with the recently developed contour-integral based eigensolvers to help them detect whether all desired eigenvalues are found. Numerical experiments are reported to show the viability of our new method.

Key words. eigenvalue, generalized eigenvalue problems, contour integral, spectral projection

AMS subject classifications. 15A18, 58C40, 65F15

1. Introduction.

Consider the generalized eigenvalue problem

$$A\mathbf{x} = \lambda B\mathbf{x}, \tag{1.1}$$

where $A, B \in \mathbb{C}^{n\times n}$. The scalars $\lambda \in \mathbb{C}$ and the associated nonzero vectors $\mathbf{x} \in \mathbb{C}^n$ are called the eigenvalues and eigenvectors, respectively. Let $\mathcal{D}$ be a disk in the complex plane and enclosed by circle Γ. In this paper, we want to develop an approach for computing the number of eigenvalues of (1.1) inside $\mathcal{D}$ exactly. Due to the Möbius transformation, the resulting approach can be adapted to the union of intersections of arbitrary half plane and (complemented) disks, and so a rather general region [3].

In many applications, it is required to know the number of eigenvalues inside a prescribed region in the complex plane [13, 14]. For example, it is a prerequisite of the eigensolvers based on divide-and-conquer techniques [2, 8]. To get this number, the most straightforward way is to compute all eigenvalues inside the target region by some method, such as the rational Krylov subspace method [17]. However, this way is always time-consuming and not effective, because we are only interested in the number of eigenvalues rather than the eigenvalues themselves.

When A and B are Hermitian matrices with B being positive definite, i.e., $zB - A$ is a definite matrix pencil [2], it is well-known that the eigenvalues of (1.1) are real-valued and lie on the real line [18, 22]. Assume that we want to know the number of eigenvalues inside interval $[a, b]$. The standard method works as follows. Compute two LDL decompositions: $A - aB = L_a D_a L_a^*$ and $A - bB = L_b D_b L_b^*$. Then the difference between the numbers of negative entries in the diagonal of D_a and D_b is exactly the number of eigenvalues inside $[a, b]$. The derivation of this method takes advantage of the Sylvester law of inertia, see [8, 12, 18] for more details. Obviously, the efficiency of this method depends on the accurate computation of two related LDL decompositions [8]. Computing the LDL decomposition requires floating point operations of order $\mathcal{O}(n^3)$ [11]; as a result, it becomes computationally prohibitive for large-scale problems. When (1.1) comes to the non-Hermitian case, to the best of our knowledge, there is no specific method for exactly computing the number of eigenvalues of (1.1) inside $\mathcal{D}$.

*Shenzhen Institutes of Advanced Technology, Chinese Academy of Science, Shenzhen, P.R. China (guojianyin@gmail.com).

The contour-integral based eigenslovers [16, 20, 21, 24] are recent efforts for computing the eigenvalues inside a prescribed region. The information about the number of eigenvalues inside the region of interest is crucial to the practical implementation of these eigensolvers. Due to this, recently some methods based on contour-integral were proposed to get estimations of this number. Some of this kind of methods make use of the stochastic estimation of the trace of spectral operator associated with the eigenvalues inside the given region [8, 9, 24]. However, they may be unreliable in some cases, for instance when the matrices A and B are ill-conditioned. In [23], another kind of contour-integral based estimate method was presented under the assumption that matrices A and B are Hermitian and B is positive definite. It should be pointed out that all these existing contour-integral based methods always just provide approximations for the exact number of eigenvalues inside the given region.

In this paper, we present a contour-integral based method, which can exactly compute the number of eigenvalues of (1.1) inside $\mathcal{D}$. The derivation of the proposed method requires the eigenvalues of (1.1) are semi-simple, namely, there are n independent eigenvectors, which is always the case in practical situations. Our new method is motivated by two findings. The first finding comes from using the Gauss-Legendre quadrature rule to approximately compute the spectral operator constructed by a particular contour integral. More details will be discussed in Section 2. The second one is devoted to avoiding the computation of the Weierstrass canonical form of matrix pencil $zB-A$ when using the first finding to count the eigenvalues inside $\mathcal{D}$. We will detail the second finding in Section 4. Since our new method is also based on contour integral, it keeps the promising features of the usual contour-integral based eigensolvers, such as having a good potential to be implemented on a high-performance parallel architecture. Moreover, it can integrate with the contour-integral based eigensolvers [16, 20, 21, 24] to help them determine whether all desired eigenvalues are found when they stop.

The paper is organized as follows. In Section 2, we present the first finding, which is derived from using the Gauss-Legendre quadrature rule to approximately compute the spectral projector constructed by an integral contour. Since our method needs the help of a technique proposed in [24], we briefly describe the technique in Section 3. In Section 4, we first detail the second finding, and then give the resulting method for counting the eigenvalues inside $\mathcal{D}$. Numerical experiments are reported in Section 5 to show the viability of our new method.

Throughout the paper, the following notation and terminology are used. The real part of a complex number a is denoted by $\Re(a)$. We use $\sqrt{-1}$ to denote the imaginary unit. The subspace spanned by the columns of matrix X is denoted by $\text{span}\{X\}$. The rank of X is denoted by $\text{rank}(X)$. The algorithms are presented in Matlab style.

2. Approximate spectral projector.

Our discussion starts with the spectral projector associated with the eigenvalues inside Γ, which is constructed by a contour integral defined as

$$Q = \frac{1}{2\pi\sqrt{-1}} \oint_{\Gamma} (zB - A)^{-1} B dz. \tag{2.1}$$

A matrix pencil $zB-A$ is called regular if $\det(zB-A)$ is not identically zero for all $z \in \mathbb{C}$ [7, 15]. Below is a generalization of the Jordan canonical form to the regular matrix pencil case.

Theorem 2.1 (Weierstrass canonical form [10]). *Let $zB-A$ be a regular matrix* *pencil of order n. Then there exist nonsingular matrices S and $T \in \mathbb{C}^{n\times n}$ such that*

$$TAS = \begin{bmatrix} J_d & 0 \\ 0 & I_{n-d} \end{bmatrix} \quad \text{and} \quad TBS = \begin{bmatrix} I_d & 0 \\ 0 & N_{n-d} \end{bmatrix}, \tag{2.2}$$

where J_d is a $d \times d$ matrix in Jordan canonical form with its diagonal entries corresponding to the eigenvalues of $zB - A$, N_{n-d} is an $(n-d) \times (n-d)$ nilpotent matrix also in Jordan canonical form, and I_d denotes the identity matrix of order d.

Assume that there are n independent eigenvectors, which implies J_d is a diagonal matrix and N_{n-d} is a zero matrix. Let $J_d = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$, with λ_i being the (finite) eigenvalues [7]. Here the λ_i are not necessarily distinct and can be repeated according to their multiplicities.

For $z \neq \lambda_i$, the matrix $(zI_d - J_d)$ is invertible. Hence, according to (2.2), the resolvent operator

$$\begin{aligned}(zB - A)^{-1}B &= S \begin{bmatrix} (zI_d - J_d)^{-1} & 0 \\ 0 & (zN_{n-d} - I_{n-d})^{-1} \end{bmatrix} TB \\ &= S \begin{bmatrix} (zI_d - J_d)^{-1} & 0 \\ 0 & (zN_{n-d} - I_{n-d})^{-1} \end{bmatrix} \begin{bmatrix} I_d & 0 \\ 0 & N_{n-d} \end{bmatrix} S^{-1} \\ &= SD(z)S^{-1}, \end{aligned} \tag{2.3}$$

where

$$D(z) = \begin{bmatrix} (zI_d - J_d)^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.4}$$

and the diagonal block $(zI_d - J_d)^{-1}$ is of the form:

$$(zI_d - J_d)^{-1} = \begin{bmatrix} \frac{1}{z-\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{z-\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{z-\lambda_d} \end{bmatrix}. \tag{2.5}$$

Assume that there are s eigenvalues enclosed by Γ, without loss of generality, let them be $\{\lambda_1, \ldots, \lambda_s\}$. Then, according to the residue theorem in complex analysis [1], it follows from (2.3)–(2.5) that

$$Q = S \left[\frac{1}{2\pi\sqrt{-1}} \oint_\Gamma D(z)dz \right] S^{-1} = S \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix} S^{-1} = S_{(:,1:s)}(S^{-1})_{(1:s,:)}. \tag{2.6}$$

It is easy to verify that $Q^2 = Q$, which implies that Q is a spectral projector onto the eigenspace $\operatorname{span}\{S_{(:,1:s)}\}$ corresponding to $\{\lambda_1, \lambda_2, \ldots, \lambda_s\}$ [24].

In view of (2.6), the spectral projector Q can be obtained via computing the Weierstrass canonical form (cf. (2.2)). However, it is well-known that the Weierstrass canonical form is not suitable for numerical computation [2, 7]. According to the expression (2.1), an alternative way is to compute Q numerically by a quadrature scheme. In our method, the quadrature scheme is restricted to the Gauss-Legendre quadrature rule [6]. Let c and ρ be the center and the radius of circle Γ, respectively. Applying the q-point Gauss-Legendre quadrature rule to (2.1) yields

$$Q \approx \widetilde{Q} = \frac{1}{2}\sum_{j=1}^{q} \omega_j (z_j - c)(z_j B - A)^{-1} B = S \left[\frac{1}{2}\sum_{j=1}^{q} \omega_j (z_j - c) D(z_j) \right] S^{-1}. \quad (2.7)$$

Here $z_j = c + \rho e^{\sqrt{-1}\theta_j}$, $\theta_j = (1 + t_j)\pi$, and t_j is the jth Gaussian node with associated weight ω_j. We remark that $\widetilde{Q}$ is an approximate spectral projector. Let

$$D = \frac{1}{2}\sum_{j=1}^{q} \omega_j (z_j - c) D(z_j). \quad (2.8)$$

Comparing to (2.6) and (2.7), we see that D is an approximation to $\begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix}$.

Let $\mu = c + re^{\sqrt{-1}\theta}$, where $r \in [0, \infty)$ and $\theta \in (-\pi, \pi]$. Define

$$\psi(\mu) = \frac{1}{2\pi\sqrt{-1}} \oint_{\Gamma} \frac{1}{z - \mu} dz. \quad (2.9)$$

According to the residue theorem, we know that $\psi(\mu) = 1$ when μ is located inside Γ, and $\psi(\mu) = 0$ when μ is located outside Γ. If $\psi(\mu)$ is computed approximately by the q-point Gauss-Legendre quadrature rule, then we have

$$\begin{aligned} \psi(\mu) \approx \widetilde{\psi}(\mu) &= \frac{1}{2}\sum_{j=1}^{q} \omega_j (z_j - c) \frac{1}{z_j - \mu} \\ &= \frac{1}{2}\sum_{j=1}^{q} \omega_j \frac{-\rho\cos(t_j\pi) - \sqrt{-1}\rho\sin(t_j\pi)}{-(\rho\cos(t_j\pi) + r\cos\theta) - \sqrt{-1}(\rho\sin(t_j\pi) + r\sin\theta)}. \end{aligned} \quad (2.10)$$

It was shown that $\widetilde{\psi}(\mu) \geqslant \frac{1}{2}$ if μ is real-valued and located inside Γ [23]. This observation sheds light on the following theorem.

Theorem 2.2. *If μ is enclosed by Γ, then the real part of $\widetilde{\psi}(\mu)$ satisfies*

$$\Re[\widetilde{\psi}(\mu)] > \frac{1}{2}.$$

If μ is located outside Γ, we have

$$\Re[\widetilde{\psi}(\mu)] < \frac{1}{2}.$$

Proof. According to (2.10), we have

$$\Re[\widetilde{\psi}(\mu)] = \frac{1}{2}\sum_{j=1}^{q} \omega_j \frac{\rho^2 + \rho r\cos(t_j\pi - \theta)}{\rho^2 + r^2 + 2\rho r\cos(t_j\pi - \theta)}. \quad (2.11)$$

Let

$$g_j(r, \theta) = \frac{\rho^2 + \rho r\cos(t_j\pi - \theta)}{\rho^2 + r^2 + 2\rho r\cos(t_j\pi - \theta)}.$$

For any given j, one can show that

$$g_j(r,\theta) - \frac{1}{2} = \frac{\rho^2 - r^2}{2\left[(\rho + r\cos(t_j\pi - \theta))^2 + (r\sin(t_j\pi - \theta))^2\right]} \tag{2.12}$$

Note that the denominator of the right hand in (2.12) is always positive. It is readily to see that

$$g_j(r,\theta) > \frac{1}{2}$$

for $r \in [0,\rho)$ and $\theta \in (-\pi,\pi]$, in which case μ is enclosed by Γ, and

$$g_j(r,\theta) < \frac{1}{2}$$

for $r \in (\rho,+\infty]$ and $\theta \in (-\pi,\pi]$, in which case μ is located outside Γ.

On the other hand, it is well-known that $\sum_{j=1}^{q} \omega_j = 2$ [6, 23]. Therefore,

$$\Re[\widetilde{\psi}(\mu)] = \frac{1}{2}\sum_{j=1}^{q} \omega_j g_j(r,\theta) > \frac{1}{2}$$

when μ is enclosed by Γ, and

$$\Re[\widetilde{\psi}(\mu)] < \frac{1}{2}$$

when μ is located outside Γ. $\square$

We use Figure 2.1 to depict the function $\Re[\widetilde{\psi}(\mu)]$. The figure well demonstrates the conclusion in Theorem 2.2.

Observe (2.4), (2.5), and (2.8)–(2.10), we see that

$$D_{(i,i)} = \widetilde{\psi}(\lambda_i), \quad i = 1,\ldots d.$$

Due to this, in the following we always refer to $D_{(i,i)}$ as the diagonal entry of D that corresponds to eigenvalue $\lambda_i, i = 1,\ldots,d$. In viewing of Theorem 2.2, the diagonal entries of D can be divided into two categories, that is, $\{\Re[D_{(i,i)}] > \frac{1}{2}\}_{i=1}^{s}$ on one group and $\{\Re[D_{(j,j)}] < \frac{1}{2}\}_{j=s+1}^{n}$ on the other. This fact is the first finding of our work, which implies that the number s can be obtained by counting the diagonal entries of D whose real parts are larger than $\frac{1}{2}$. According to (2.8), getting D requires to compute the Weierstrass canonical form of $zB - A$. However, as was suggested in [2], the Weierstrass canonical form is not suitable for numerical computation.

In our work, we present an alternative method to obtain $\{D_{(i,i)}\}_{i=1}^{s}$, which does not need to compute the Weierstrass canonical form of $zB - A$. As a result, by exploiting the first finding, we can exactly compute the number of eigenvalues of (1.1) inside Γ. The alternative method is our second finding, since it needs the help of a technique proposed in [24], we briefly introduce the technique in next section before presenting the second finding.

3. Finding an upper bound for the number of eigenvalues inside Γ.

In [24], a method based on contour integral was proposed for finding a good upper bound for the number of eigenvalues inside a given region. Meanwhile, it can produce an approximate projection onto the eigenspace corresponding to the eigenvalues inside the given region.

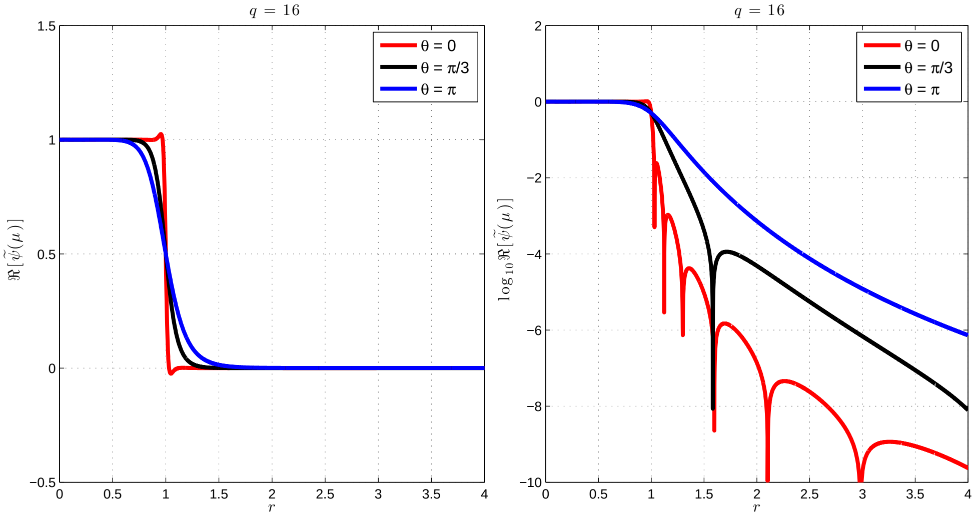

FIG. 2.1. *This figure illustrates the function $\Re[\widetilde{\psi}(\mu)]$ (cf. (2.11)) when $r \in [0, 4]$. The circle Γ has radius $\rho = 1$ with center at the origin. We set the degree of Gauss-Legendre quadrature rule $q = 16$. The left picture shows the general shape of the function, and the right one shows the logarithmic scale.*

The method first uses a stochastic estimation of the trace of spectral projector Q (cf. (2.1)) to obtain an estimation of s. By $Y_p \sim \mathsf{N}_{n\times p}(0,1)$, we mean that Y_p is an $n \times p$ random matrix with independent and identically distributed entries drawing from standard normal distribution $\mathsf{N}(0,1)$. By (2.6), one can easily verify that

$$\begin{aligned}\frac{1}{p}\mathbb{E}[\mathrm{trace}(Y_p^* Q Y_p)] &= \mathrm{trace}(Q) = \mathrm{trace}(S_{(:,1:s)}(S^{-1})_{(1:s,:)})\\ &= \mathrm{trace}((S^{-1})_{(1:s,:)}S_{(:,1:s)}) = \mathrm{trace}(I_s) = s.\end{aligned}$$

Therefore,

$$s_0 := \lceil \frac{1}{p}\mathrm{trace}(Y_p^* Q Y_p)\rceil \tag{3.1}$$

provides an estimation for s, see [9, 19, 24] for more details. With this knowledge on hand, a method was then given in [24] to seek a good upper bound s_1 of s. Wanting to derive the method, we need the following lemma.

LEMMA 3.1 ([24]). *Let $Y \in \mathbb{R}^{n\times m}$. If the entries of Y are continuous random numbers that are independent and identically distributed (i.i.d.), then the matrix $(S^{-1})_{(1:m,:)}Y$ is almost surely nonsingular.*

Let $s^\dagger$ be a positive integer and $Y_{s^\dagger} \sim \mathsf{N}_{n\times s^\dagger}(0,1)$. Consider

$$U_{s^\dagger} = QY_{s^\dagger} = \frac{1}{2\pi\sqrt{-1}}\oint_\Gamma (zB-A)^{-1}B dz Y_{s^\dagger} = S_{(:,1:s)}(S^{-1})_{(1:s,:)}Y_{s^\dagger}.$$

Hence $U_{s^\dagger}$ is the projection of $Y_{s^\dagger}$ onto $\mathrm{span}\{S_{(:,1:s)}\}$, which implies $\mathrm{rank}(U_{s^\dagger}) \le s$. With this in mind, if $\mathrm{rank}(U_{s^\dagger}) = s^\dagger$, then we have $s^\dagger \le s$. Otherwise, if $\mathrm{rank}(U_{s^\dagger}) < s^\dagger$, we can conclude that $s = \mathrm{rank}(U_{s^\dagger})$ with the help of Lemma 3.1, and thereby

$s < s^\dagger$. Based on these arguments, the following algorithm was proposed in [24] for finding a good upper bound for s. Meanwhile, a projection matrix onto the eigenspace $\mathrm{span}\{S_{(:,1:s)}\}$ is produced, which will play an important role in the resulting method given in next section.

Algorithm 1. *Input an increasing factor $\alpha > 1$ and the size p of sample vectors. The function "Search" outputs s_1, an upper bound of the number of eigenvalues s inside Γ, and a projection matrix $U_1 \in \mathbb{C}^{n\times s_1}$ onto* $\mathrm{span}\{S_{(:,1:s)}\}$.

```
Function $[U_1, s_1] = \textsc{Search}(A, B, \Gamma, \alpha, p)$
1.  Pick $Y_p \sim \mathsf{N}_{n\times p}(0,1)$ and compute $U = \frac{1}{2\pi\sqrt{-1}}\oint_\Gamma (zB-A)^{-1}BdzY_p$
    by the $q$-point Gauss-Legendre quadrature rule.
2.  Set $s_0 = \lceil \frac{1}{p}\mathrm{trace}(Y_p^* U)\rceil$ and $s^\star = \min(\max(p, s_0), n)$.
3.  If $s^\star > p$
4.      Pick $\hat{Y} \sim \mathsf{N}_{n\times(s^\star-p)}(0,1)$ and compute $\hat{U} = \frac{1}{2\pi\sqrt{-1}}\oint_\Gamma (zB-A)^{-1}Bdz\hat{Y}$
        by the $q$-point Gauss-Legendre quadrature rule.
5.      Augment $\hat{U}$ to $U$ to form $U = [U, \hat{U}] \in \mathbb{C}^{n\times s^\star}$.
6.  Else
7.      Set $s^\star = p$.
8.  End
9.  Compute $U\Pi = U_1R_1$: the rank-revealing QR decomposition [5] of $U$.
10. Set $s_1 = \mathrm{rank}(R)$.
11. If $s_1 < s^\star$, stop. Otherwise, set $p = s_1$ and $s^\star = \lceil \alpha s_1\rceil$. Then go to Step 3.
```

In practice, by (2.7), we see that U (line 5) formed in the last iteration in Algorithm 1 is

$$\begin{aligned} U = \widetilde{Q}Y_{s^\star} &= S\left[\frac{1}{2}\sum_{j=1}^{q}\omega_j(z_j-c)D(z_j)\right]S^{-1}Y_{s^\star} = SDS^{-1}Y_{s^\star} \\ &= [D_{(1,1)}S_{(:,1)}, \ldots, D_{(d,d)}S_{(:,d)}, 0, \ldots, 0]S^{-1}Y_{s^\star}, \end{aligned} \tag{3.2}$$

where $Y_{s^\star} \sim \mathsf{N}_{n\times s^\star}(0,1)$ with $s^\star > s_1$. According to Lemma 3.1, we know that $S^{-1}Y_{s^\star}$ is full-rank, then it follows from (3.2) that

$$\mathrm{span}\{U_1\} = \mathrm{span}\{U\} = \mathrm{span}\{S_{(:,\mathcal{I})}\}, \tag{3.3}$$

where $\mathcal{I}$ is an index set and its cardinality is s_1. It has been shown in previous section that $\{\Re[D_{(i,i)}] > \frac{1}{2}\}_{i=1}^{s}$, thus we can conclude that $\{1, 2, \ldots, s\} \subset \mathcal{I}$.

4. Counting the eigenvalues inside Γ.

In this section, we present the resulting method for counting the eigenvalues inside a given disk in the complex plane.

Below is the second finding of our work.

Theorem 4.1. *Let U_1 be the projection matrix computed by Algorithm 1. Let*

$$U_2 = QU_1 = \frac{1}{2\pi\sqrt{-1}}\oint_\Gamma (zB-A)^{-1}BdzU_1.$$

Compute U_2 by the q-point Gauss-Legendre quadrature rule, i.e.,

$$U_2 \approx \widetilde{U}_2 = \widetilde{Q}U_1 = \frac{1}{2}\sum_{j=1}^{q}\omega_j(z_j-c)(z_jB-A)^{-1}BU_1, \tag{4.1}$$

and define the $s_1 \times s_1$ *matrix*

$$M = U_1^* \widetilde{U}_2. \tag{4.2}$$

Then $\{D_{(i,i)}\}_{i=1}^s$ *are the eigenvalues of* M.

Proof. Since U_1 is computed by Algorithm 1, by (3.3), we can find a nonsingular matrix, say, W such that

$$S_{(:,\mathcal{I})} = U_1 W.$$

Let $\bar{\mathcal{I}}$ be the index set $\{1, 2, \ldots, n\} \setminus \mathcal{I}$, then there exists a permutation matrix P such that

$$SP = [S_{(:,\mathcal{I})}, S_{(:,\bar{\mathcal{I}})}].$$

Note that D is a diagonal matrix, thereby P^*DP is also a diagonal matrix and can be written as

$$P^*DP = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix},$$

where the diagonal entries of D_1 consist of $D_{(i,i)}, i \in \mathcal{I}$.

By (2.7), (4.1) and (4.2), we have

$$\begin{aligned} M &= U_1^* \widetilde{U}_2 = U_1^*(SDS^{-1})U_1 \\ &= U_1^*(SPP^*DPP^*S^{-1})U_1 \\ &= U_1^*[S_{(:,\mathcal{I})} D_1 (S^{-1})_{(\mathcal{I},:)} + S_{(:,\bar{\mathcal{I}})} D_2 (S^{-1})_{(\bar{\mathcal{I}},:)}](S_{(:,\mathcal{I})} W^{-1}) \\ &= U_1^* S_{(:,\mathcal{I})} D_1 W^{-1} \\ &= U_1^*(U_1 W) D_1 W^{-1} \\ &= W D_1 W^{-1}. \end{aligned}$$

Since W is nonsingular, the matrices M and D_1 have the same eigenvalues, which are $\{D_{(i,i)}\}_{i \in \mathcal{I}}$. Note that $\{1, 2, \ldots, s\} \subset \mathcal{I}$, therefore $\{D_{(i,i)}\}_{i=1}^s$ are the eigenvalues of M. □

Theorem 4.1 tells us that $\{D_{(i,i)}\}_{i=1}^s$ can be obtained via computing the eigenvalues of M (cf. (4.2)). We have shown in Section 2 that the real parts of $\{D_{(i,i)}\}_{i=1}^s$, which correspond to the eigenvalues inside Γ, are larger than $\frac{1}{2}$, and the real parts of the rest diagonal entries of D are smaller than $\frac{1}{2}$. Motivated by these facts, we find the number s via counting the eigenvalues of M whose real parts are larger than $\frac{1}{2}$. Summarize this idea, below we give the complete algorithm for computing the number of eigenvalues inside Γ.

ALGORITHM 2. *Input an increasing factor* $\alpha > 1$ *and the size* p *of sample vectors. The function "*COUNT_EIGS*" computes the number of eigenvalues of (1.1) that are located inside circle* Γ.

```
Function $s$ = COUNT_EIGS($A, B, \Gamma, \alpha, p$)
1. Call $[U_1, s_1]$ = SEARCH($A, B, \Gamma, \alpha, p$).
2. Compute $\widetilde{U}_2$ in (4.1), and set $M = U_1^* \widetilde{U}_2$.
3. Compute the eigenvalues of $M$, and set $s$ to be the number of the computed
   eigenvalues whose real parts are larger than $\frac{1}{2}$.
```

As with other contour-integral based eigensolvers [16, 20, 21, 24], the dominant work of Algorithm 2 is solving generalized shifted linear systems of form

$$(z_j B - A)X_j = BY, \tag{4.3}$$

which can be solved by any method of choice. Since the quadrature nodes z_j and the columns of the right-hand sides of (4.3) are independently, Algorithm 2 has good scalability in modern parallel architectures.

The information about the number of eigenvalues inside the target region is crucial to the implementation of the contour-integral based eigensolvers. Algorithm 2 can integrate with this kind of eigensolvers [16, 20, 21, 24] to provide them with this important information. For example, if we use the contour-integral based eigensolver proposed in [24] to compute the eigenvalues inside Γ, U_1 and $\widetilde{U}_2$ need to be computed in the first iteration and the second, respectively. As a result, to get s, the extra work is constructing the matrix M (cf. (4.2)) and then computing the eigenvalues of M. Once the number s is obtained, it can help to detect whether all eigenvalues inside Γ are found in the subsequent iterations and determine when to stop the iteration process. For clarity, we summarize the idea of integrating Algorithm 2 with the eigensolver proposed in [24] by the following algorithm.

Algorithm 3. *Input $A, B \in \mathbb{C}^{n\times n}$, an increasing factor $\alpha > 1$, the size p of sample vectors, a circle Γ, a convergence tolerance ϵ, and "max_iter" to control the maximum number of iterations. The function "Eigenpairs" computes eigenpairs $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ of (1.1) that satisfies*

$$\tilde{\lambda}_i \text{ inside } \Gamma \quad \text{and} \quad \frac{\|A\tilde{\mathbf{x}}_i - \tilde{\lambda}_i B\tilde{\mathbf{x}}_i\|_2}{\|A\tilde{\mathbf{x}}_i\|_2 + \|B\tilde{\mathbf{x}}_i\|_2} < \epsilon. \tag{4.4}$$

The results are stored in vector Λ and matrix X.

```
Function $[\Lambda, X] = \textsc{Eigenpairs}(A, B, \Gamma, \epsilon, max\_iter)$
1.  Call $[U_1, s_1] = \textsc{Search}(A, B, \Gamma, \alpha, p)$.
2.  Compute $\widetilde{U}_2$ in (4.1), and let $M = U_1^* \widetilde{U}_2$. Set $s$ to be the number of the
    eigenvalues of $M$ whose real parts are larger than $\frac{1}{2}$.
3.  For $k = 2, \cdots, max\_iter$
4.      Compute QR decompositions: $\widetilde{U}_k = U_1 R_1$ and $B\widetilde{U}_k = U_2 R_2$.
5.      Form $\widetilde{A} = U_2^* A U_1$ and $\widetilde{B} = U_2^* B U_1$.
6.      Solve the projected eigenproblem $\widetilde{A}\mathbf{y} = \tilde{\lambda}\widetilde{B}\mathbf{y}$ of size $s_1$ to obtain eigenpairs
        $\{(\tilde{\lambda}_i, \mathbf{y}_i)\}_{i=1}^{s_1}$. Set $\tilde{\mathbf{x}}_i = U_1 \mathbf{y}_i, i = 1, 2, \ldots, s_1$.
7.      Set $\Lambda = [\ ]$ and $X = [\ ]$.
8.      For $i = 1 : s_1$
9.          If $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i)$ satisfies (4.4), then $\Lambda = [\Lambda, \tilde{\lambda}_i]$ and $X = [X, \tilde{\mathbf{x}}_i]$.
10.     End
11.     If there are $s$ eigenpairs satisfying (4.4), stop. Otherwise, compute
        $U_{k+1} = \frac{1}{2\pi\sqrt{-1}} \oint_\Gamma (zB - A)^{-1} B dz U_1$ by the $q$-point Gauss-Legendre
        quadrature rule: $U_{k+1} \approx \widetilde{U}_{k+1} = \frac{1}{2} \sum_{j=1}^{q} \omega_j (z_j - c)(z_j B - A)^{-1} B U_1$.
12. End
```

Below we give some remarks on above algorithm.

1. The first two steps can be viewed as using Algorithm 2 to determine the number of eigenvalues inside Γ. In addition, a projection matrix is also produced, i.e., $\widetilde{U}_2$.

2. Since $\widetilde{U}_2$ is a projection matrix onto span$\{S_{(:,1:s)}\}$, it can be used to construct a projected eigenproblem to compute approximate eigenpairs. Therefore, in Step 3, the for-loop starts from $k = 2$.
3. In Step 11, the number s computed in Step 2 helps to detect whether all s approximate eigenpairs satisfying the prescribed accuracy. If it is, we stop the iteration process.

5. Numerical Experiments. In this section, we give some numerical experiments to illustrate the viability of our new method. All computations are carried out in MATLAB version R2012b on a MacBook with an Intel Core i5 2.5 GHz processor and 8 GB RAM.

In the experiments, as for computing the generalized shifted linear systems of the form (4.3), we first use the MATLAB function `lu` to compute the LU decomposition of $z_jB - A, j = 1, 2, \dots, q$, and then perform the triangular substitutions to get the corresponding solutions.

Experiment 5.1: Our method (Algorithm 2) is motivated by two findings: (i) for matrix D (cf. (2.8)), $\Re[D_{(i,i)}] > \frac{1}{2}$ if $D_{(i,i)}$ correspond to the eigenvalues enclosed by Γ, and $\Re[D_{(i,i)}]$ are less than $\frac{1}{2}$ if $D_{(i,i)}$ correspond to the eigenvalues outside Γ; (ii) $D_{(i,i)}$ are the eigenvalues of M (cf. (4.2)) if they correspond to the eigenvalues within Γ. This experiment is devoted to illustrating these two findings.

Let $\Lambda = \texttt{diag}([0.1 : 0.1 : 0.8])$, $S = \texttt{randn}(8)$, the matrices A and B are given by

$$A = S\Lambda S^{-1}, \qquad B = \texttt{eye}(8).$$

Here `diag`, `randn`, and `eye` are MATLAB commands. Obviously, for the problem under consideration, the eigenvalues are $0.1, 0.2, \dots, 0.8$. Let Γ be a circle with center at origin and radius $\rho = 0.401$. Suppose that we are interested in the number of eigenvalues inside Γ. Obviously, there are 4 eigenvalues are located inside Γ. Note that the eigenvalue 0.4 is located inside Γ and close to boundary of the disk surrounded by Γ.

In this experiment, we select the number of quadrature points $q = 32$. According to (2.8), now D is given by

$$D = \sum_{i=1}^{32} \omega_i z_i (z_i I_8 - \Lambda)^{-1}, \tag{5.1}$$

where ω_i are the weights associated with quadrature nodes z_i. We take the size of sample vectors $p = 6$, thus the starting basis in function SEARCH is $Y_6 = \texttt{randn}(8, 6)$. Since the size of test problem is small and the number of columns of Y_6 is already larger than the number of eigenvalues inside Γ, we just run one iteration when preforming function SEARCH to get projection matrix U_1. As a result, the matrix M defined in (4.2) is of size 6×6.

Since, in practical situations, we are only interested in the real parts of the diagonal entries of D, in the second column in TABLE 5.1 we list $\{\Re[D_{(i,i)}]\}_{i=1}^{8}$ that are computed by (5.1). It can be seen that the $\{\Re[D_{(i,i)}]\}_{i=1}^{4}$, corresponding to the eigenvalues inside Γ, are larger than 0.5. More precisely, $\Re[D_{(i,i)}], i = 1, 2, 3$, corresponding to eigenvalues 0.1, 0.2, 0.3, respectively, approximate the theoretical value 1 sufficiently. But $\Re[D_{(4,4)}]$ is about 0.8, this is because it corresponds to the eigenvalue 0.4, which is close to the boundary of the target disk. On the other hand, the real parts of $\{D_{(i,i)}\}_{i=5}^{8}$, corresponding to the eigenvalues outside Γ, are less than 0.5 and very close to zero, as expected of our first finding.

Table 5.1

The real parts of the diagonal entries of D and the ones of the eigenvalues of M.

i	$\Re[(D_{(i,i)})]$	$\Re[\texttt{eig}(M)]$
1	$\underline{1.0000000000039}49$	$\underline{1.0000000000039}65$
2	$\underline{1.0000000000000}00$	$\underline{1.0000000000000}12$
3	$\underline{0.999999999999999}$	$\underline{0.999999999999999}$
4	$\underline{0.8015817876596}01$	$\underline{0.8015817876596}10$
5	$\underline{0.0000000025256}84$	$\underline{0.0000000025256}20$
6	$\underline{0.0000000000043}79$	$\underline{0.0000000000043}80$
7	0.000000000000001	
8	−0.000000000000051	

Table 5.2

Test problems from Matrix Market that are used in Experiment 5.2

No.	Matrix	Size	nnz	Property	`cond`
1	A: BFW398A	398	3678	unsymmetric	7.58×10^3
	B: BFW398B	398	2910	symmetric indefinite	3.64×10^1
2	A: BFW782A	782	7514	unsymmetric	4.63×10^3
	B: BFW782B	782	5982	symmetric indefinite	3.05×10^1
3	A: BCSSTK08	1074	12960	symmetric positive definite	4.77×10^7
	B: BCSSTM08	1074	1074	symmetric positive definite	8.27×10^6
4	A: BCSSTK27	1224	28675	symmetric positive definite	7.71×10^4
	B: BCSSTM27	1224	28675	symmetric indefinite	1.14×10^{10}
5	A: PLAT1919	1919	17159	symmetric indefinite	1.40×10^{16}
	B: PLSK1919	1919	4831	skew symmetric	1.07×10^{18}
6	A: BCSSTK13	2003	42943	symmetric positive definite	4.57×10^{10}
	B: BCSSTM13	2003	11973	symmetric positive semi-definite	Inf
7	A: MHD4800A	4800	102252	unsymmetric	2.54×10^{57}
	B: MHD4800B	4800	27520	symmetric indefinite	1.03×10^{14}
8	A: BCSSTK25	15439	252241	symmetric indefinite	1.28×10^{13}
	B: BCSSTM25	15439	15439	symmetric positive definite	6.06×10^9

The third column in Table 5.1 displays the real parts of the eigenvalues of M in descent order. The same digits of $\Re[D_{(i,i)}]$ and the ith largest $\Re[\texttt{eig}(M)]$, $i = 1, \ldots, 6$ are underlined. We can see that $\Re[D_{(i,i)}], i = 1, \ldots, 6$, agree at least fourteen digits to their counterparts in the third column. Therefore the real parts of the eigenvalues of M are almost equivalent to $\Re[D_{(i,i)}], i = 1, \ldots, 6$, which justifies the second finding of our work.

Experiment 5.2: This experiment is devoted to testing the viability of our new method. The test matrices are available from the Matrix Market collection [4]. They are the real-world problems from scientific and engineering applications. The descriptions of the matrices are presented in Table 5.2, where nnz denotes the number of non-zero entries and their condition numbers are computed by Matlab function `condest`. The test matrices of different problems vary in size, spectrum and property.

In this experiment, we take the parameter q to be 16. The parameters c and ρ are the center and the radius of circle Γ, respectively. Note that the test problems 3, 4, 6, and 8 are Hermitian problems, which means their (finite) eigenvalues are real-valued. Due to this, we choose the circles with centers lying on the real line for these test problems.

Table 5.3 presents the numerical comparisons. s is the actual number of eigenvalues inside Γ. We first use the Matlab built-in function `eig` to compute all eigenvalues of the test problems, and then determine the values of s according to the coordinates of computed eigenvalues. We also present the estimation s_0 computed by the trace

TABLE 5.3

Numerical comparison: s is the exact number of eigenvalues inside Γ, s_0 is the estimate of s by using the trace formula and s_1 is the upper bound computed by Algorithm 1, and Cont_Eigs is the result computed by our new method.

No.	(c, ρ)	s	s_0	s_1	Cont_Eigs
1	$((-6.0 \times 10^5) + \sqrt{-1}(2.0 \times 10^5), 3.0 \times 10^5)$	120	136	177	120
2	$((-5.0 \times 10^5) + \sqrt{-1}(1.0 \times 10^5), 2.0 \times 10^5)$	165	161	228	165
3	$(5.0 \times 10^5, 3.0 \times 10^5)$	178	199	190	178
4	$(5.0 \times 10^3, 3.0 \times 10^3)$	160	134	192	160
5	$(\sqrt{-1}(5.0 \times 10^{-1}), 1.5 \times 10^{-1})$	301	297	397	301
6	$(6.0 \times 10^6, 3.5 \times 10^6)$	232	229	262	232
7	$((-5.0 \times 10^2) + \sqrt{-1}(2.0 \times 10^2), 4.0 \times 10^2)$	212	545	293	212
8	$(5.0 \times 10^5, 2.0 \times 10^5)$	1663	1628	1749	1663

formula (3.1) and the upper bound s_1 computed by Algorithm 1. Cont_Eigs is the result computed by our new method.

The results for all eight test problems are reported in TABLE 5.3. From these data, we see that the estimation s_0 computed by the trace formula (3.1) always provide good estimation to s for all test problems except for test problem 7, due to its ill-condition. s_1 always gives a good upper bound for s. It is remarkable that the result computed by our new method is the same with the exact number s for each test problem, even though the test problem 7. Therefore, our new method is numerically efficient and reliable.

6. Conclusion.

In this work, we develop an approach for counting the eigenvalues of (1.1) inside a given disk in the complex plane. The new method is a contour-integral based method and motivated by two findings. The computational advantage of the new method is that it is easily parallelizable. Its another promising feature is that it can integrate with the recently proposed contour-integral based eigensolvers to provide them the information of the number of eigenvalues inside the target region. How to adapt the resulting method to the nonlinear problems will be our future work.

REFERENCES

[1] L. AHLFORS, *Complex Analysis*, 3rd Edition, McGraw-Hill, Inc., 1979.

[2] Z. BAI, J. DEMMEL, J. DONGARRA, A. RUHE, AND H. VAN DER VORST, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide* , SIAM, Philadelphia, 2000.

[3] Z. BAI, J. DEMMEL, AND M. GU, *An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblem*, Numer. Math., 76 (1997), pp. 279–308.

[4] R. F. BOISVERT, R. POZO, K. REMINGTON, R. BARRETT, AND J. DONGARRA, *The matrix market: A web repository for test matrix data*, in The Quality of Numerical Software, Assessment and Enhancement, R. Boisvert, ed., Chapman & Hall, London, 1997, pp. 125 –137.

[5] T. T. CHAN, *Rank revealing QR factorizations*, Lin. Alg. Appl., 88-89 (1987), pp. 67–82.

[6] P. J. DAVIS AND P. RABINOWITZ, *Methods of numerical integration*, 2nd Edition, Academic Press, Orlando, FL, 1984.

[7] J. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[8] E. DI NAPOLI, E. POLIZZI, AND Y. SAAD, *Efficient estimation of eigenvalue counts in an interval*, http://arxiv.org/abs/1308.4275.

[9] Y. FUTAMURA, H. TADANO, AND T. SAKURAI, *Parallel stochastic estimation method of eigenvalue distribution*, JSIAM Letters 2 (2010), pp.127–130.

[10] F. R. GANTMACHER, *The Theory of Matrices*, Chelsea, New York, 1959.

[11] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd Edition, Johns Hopkins University Press, Baltimore, MD, 1996.

[12] R. G. Grimes, J. D. Lewis, and H. D. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.

[13] E. R. Kamgnia and B. Philippe, *Counting eigenvalues in domains of the complex field*, http://arxiv.org/abs/1110.4797.

[14] L. Lin, Y. Saad, and C. Yang, *Approximating spectral densities of large matrices*, http://arxiv.org/abs/1308.5467.

[15] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[16] E. Polizzi, *Density-matrix-based algorithm for solving eigenvalue problems*, Phys. Rev. B 79 (2009) 115112.

[17] A. Ruhe, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551.

[18] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, SIAM, Philadelphia, 2011.

[19] T. Sakurai, Y. Futamura, and H. Tadano, *Efficient parameter estimation and implementation of a contour integral-based eigensolver*, J. Alg. Comput. Tech., 7 (2014), pp. 249–269.

[20] T. Sakurai and H. Sugiura, *A projection method for generalized eigenvalue problems using numerical integration*, J. comput. Appl. Math., 159 (2003), pp. 119–128.

[21] T. Sakurai and H. Tadano, *CIRR: A Rayleigh–Ritz type method with contour integral for generalized eigenvalue problems*, Hokkaido Math. J. 36 (2007), pp. 745–757.

[22] G. W. Stewart, *Matrix Algorithms, Vol. II, Eigensystems*, SIAM, Philadelphia, 2001.

[23] P. T. 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.

[24] G. Yin, R. Chan, and M-C. Yueng, *A FEAST algorithm for generalized non-Hermitian eigenvalue problems*, http://arxiv-web3.library.cornell.edu/abs/1404.1768.