FAST COMPUTATION OF SPECTRAL DENSITIES FOR GENERALIZED EIGENVALUE PROBLEMS

YUANZHE XI *, RUIPENG LI †, AND YOUSEF SAAD *

Abstract. The distribution of the eigenvalues of a Hermitian matrix (or of a Hermitian matrix pencil) reveals important features of the underlying problem, whether a Hamiltonian system in physics, or a social network in behavioral sciences. However, computing all the eigenvalues explicitly is prohibitively expensive for real-world applications. This paper presents two types of methods to efficiently estimate the spectral density of a matrix pencil (A, B) when both A and B are Hermitian and, in addition, B is positive definite. The first one is based on the Kernel Polynomial Method (KPM) and the second on Gaussian quadrature by the Lanczos procedure. By employing Chebyshev polynomial approximation techniques, we can avoid direct factorizations in both methods, making the resulting algorithms suitable for large matrices. Under some assumptions, we prove bounds that suggest that the Lanczos method converges twice as fast as the KPM method. Numerical examples further indicate that the Lanczos method can provide more accurate spectral densities when the eigenvalue distribution is highly non-uniform. As an application, we show how to use the computed spectral density to partition the spectrum into intervals that contain roughly the same number of eigenvalues. This procedure, which makes it possible to compute the spectrum by parts, is a key ingredient in the new breed of eigensolvers that exploit "spectrum slicing".

Key words. Spectral density, density of states, generalized eigenvalue problems, spectrum slicing, Chebyshev approximation, perturbation theory.

AMS subject classifications. 15A18, 65F10, 65F15, 65F50

1. Introduction.

The problem of estimating the *spectral density* of an $n \times n$ Hermitian matrix A, has many applications in science and engineering. The spectral density is termed *density of states* (DOS) in solid state physics where it plays a key role. Formally, the DOS is defined as

$$\phi(t) = \frac{1}{n} \sum_{j=1}^{n} \delta(t - \lambda_j), \tag{1.1}$$

where δ is the Dirac δ-function or Dirac distribution, and the λ_j's are the eigenvalues of A, assumed here to be labeled increasingly. In general, the formal definition of the spectral density as expressed by (1.1) is not easy to use in practice. Instead, it is often approximated, or more specifically smoothed, and it is this resulting approximation, usually a smooth function, that is sought.

Estimating spectral densities can be useful in a wide range of applications apart from the important ones in physics, chemistry and network analysis, see, e.g., [6, 8, 20]. One such application is the problem of estimating the number $\eta_{[a,\ b]}$ of eigenvalues in an interval $[a,\ b]$. Indeed, this number can be obtained by integrating the spectral density in the interval:

$$\eta_{[a,\ b]} = \int_a^b \sum_j \delta(t - \lambda_j)\ dt \equiv \int_a^b n\phi(t) dt\ . \tag{1.2}$$

*{yxi,saad}@umn.edu; Work supported in part (applications and practical aspects) by by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences DE-SC0008877 and in part (theory) by NSF under grant CCF-1505970

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P. O. Box 808, L-561, Livermore, CA 94551 (li50@llnl.gov). This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-xxxxxx).

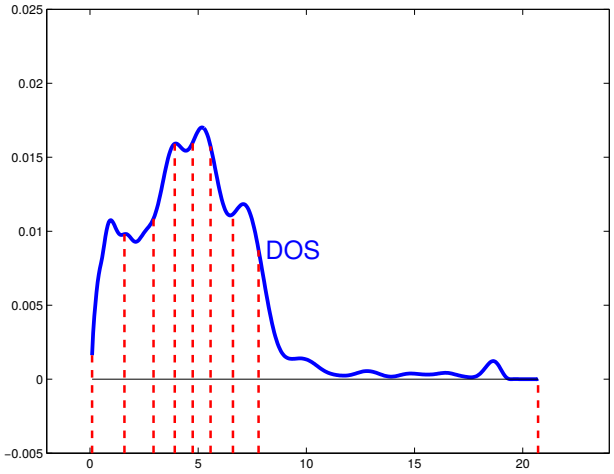


FIG. 1.1. *An illustration of slicing a spectrum into* 8 *subintervals* $[t_i,\ t_{i+1}]$ $(i = 0, \ldots, 7)$. *The solid blue curve represents a smoothed density of states (DOS) and the dotted red lines separate the subintervals.*

Thus, one can view $\phi(t)$ as a probability distribution function which gives the probability of finding eigenvalues of A in a given infinitesimal interval near t and a simple look at the DOS plot provides a sort of sketch view of the spectrum of A.

Another, somewhat related, use of density of states is in helping deploy spectrum slicing strategies [16, 18, 37]. The goal of such strategies is to subdivide a given interval of the spectrum into subintervals in order to compute the eigenvalues in each subinterval independently. Note that this is often done to balance memory usage rather than computational load. Indeed, load balancing cannot be assured by just having slices with roughly equal numbers of eigenvalues. With the availability of the spectral density function ϕ, slicing the spectrum contained in an interval $[a,\ b]$ into n_s subintervals can be easily accomplished. Indeed, it suffices to find intervals $[t_i,\ t_{i+1}]$, $i = 0, \cdots, n_s - 1$, with $t_0 = a$ and $t_{n_s} = b$ such that

$$\int_{t_i}^{t_{i+1}} \phi(t)dt = \frac{1}{n_s}\int_a^b \phi(t)dt, \quad i = 0, 1, \cdots, n_s - 1.$$

See Fig. 1.1 for an illustration and Section 5.3 for more details.

A non-standard and important use of spectral densities is when estimating numerical ranks of matrices [32, 33]. In many applications, a given $m \times n$ data matrix A (say with $m > n$) is known to correspond to a phenomenon that should yield vectors lying in a low-dimensional space. With noise and approximations the resulting data is no longer of low-rank but it may be nearly low-rank in that its numerical rank is small. It may be important in these applications to obtain this numerical rank. In [32, 33] the authors developed a few heuristics that exploit the spectral density for this task. The main idea is that for a nearly low-rank matrix, the spectral density should be quite high near the origin of the matrix $A^T A$ and it should drop quickly before increasing again. The numerical rank corresponds to the point when ϕ starts increasing again, i.e., when the derivative of the DOS changes signs. This simple strategy provides an efficient way to estimate the rank.

A straightforward way to obtain the spectral density of a given matrix A is to compute all its eigenvalues but this approach is expensive for large matrices. Effective alternatives based on stochastic arguments have been developed, see, [20] for a survey. Essentially all the methods described in the literature to compute the DOS rely on performing a number of products of the matrix A with random vectors. For sparse matrices or dense structured matrices with almost linear complexity matrix-vector products [4, 13], these products are inexpensive and so a fairly good approximation of the DOS can be obtained at a very low cost. On the other hand, not much work has been done to address the same problem for generalized eigenvalue problems

$$Ax = \lambda Bx. \tag{1.3}$$

This paper focuses on this specific issue as well as on the related problem on implementing spectrum slicing techniques [18, 24]. From a theoretical viewpoint the problem may appear to be a trivial extension of the standard case. However, from a practical viewpoint several difficulties emerge, e.g., it is now necessary to solve a linear system with B (or A) each time we operate on vectors in the stochastic sampling procedure or in a Lanczos procedure. For large-scale problems discretized from 3D models, factorizing B (or A) tends to be prohibitively expensive and so this naturally leads to the question: *Is it possible to completely avoid factorizations when computing the density of states for (1.3)?* As will be seen the answer is yes, i.e., it is possible to get the DOS accurately without any factorizations and at a cost that is comparable with that of standard problems in many applications. For example, the matrix B is often the mass matrix in discretizations such as the Finite Element Method (FEM). An important observation that is often made regarding these matrices is that they are strongly diagonally dominant.

In the remainder of the paper we will assume that A and B are Hermitian while, in addition, B is positive definite. We will call λ_j, $j = 1, 2, \cdots, n$ the eigenvalues of the pencil (A, B), and assume that they are labeled increasingly. We also denote by u_j the eigenvector corresponding to λ_j, so if $U = [u_1, u_2, \cdots, u_n]$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, then the pencil (A, B) admits the eigen-decomposition

$$U^T A U = \Lambda \tag{1.4}$$

$$U^T B U = I\ . \tag{1.5}$$

The rest of the paper is organized as follows. Section 2 discusses a few techniques to avoid direct factorizations when extending standard approaches for computing the DOS to the generalized eigenvalue problem. Section 3 presents the extension of the classical Kernel Polynomial Method (KPM) and Section 4 studies the Lanczos method from the angle of quadrature. We provide some numerical examples in Section 5 and draw some concluding remarks in Section 6.

2. Symmetrizing the generalized eigenvalue problem. A common way to express the generalized eigenvalue problem (1.3) is to multiply through by B^{-1}:

$$B^{-1}Ax = \lambda x. \tag{2.1}$$

This is now in the standard form but the matrix involved is non-Hermitian. However, as is well-known, the matrix $B^{-1}A$ is self-adjoint with respect to the B-inner product and this observation allows one to use standard methods, such as the Lanczos algorithm, that are designed for Hermitian matrices.

Another way to extend standard approaches for computing the spectral density is to transform the problem (1.3) into a standard one via the Cholesky factorization. First, assume that the Cholesky factorization of B is available and let it be written as $B = LL^{\mathsf{T}}$. Then the original problem (1.3) can also be rewritten as

$$L^{-1}AL^{-\mathsf{T}}y = \lambda y, \quad \text{with} \quad y = L^{\mathsf{T}}x, \tag{2.2}$$

which takes the standard form with a Hermitian coefficient matrix. This allows us to express the density of states from that of a standard problem. This straightforward solution faces a number of issues. Foremost among these is the fact that the Cholesky factorization may not be available or that it may be too expensive to compute. In the case of FEM methods, the factorization of B may be too costly for $3D$ problems.

Note that the matrix square root factorization can also be used in the same way. Here the original problem (1.3) is transformed into the equivalent problem:

$$B^{-1/2}AB^{-1/2}y = \lambda y, \quad \text{with} \quad y = B^{1/2}x, \tag{2.3}$$

which also assumes the standard form with a Hermitian coefficient matrix. The square root factorization is usually expensive to compute and may appear to be impractical at first. However, in the common situation mentioned above where B is strongly diagonally dominant, the action of $B^{-1/2}$ as well B^{-1} on a vector can be easily approximated by the matrix-vector product associated with a low degree polynomial in B. This is discussed next.

2.1. Approximating actions of B^{-1} and $B^{-1/2}$ on vectors. As was seen above computing the DOS for a pair of matrices requires matrix-vector products with either $B^{-1}A$, or $L^{-1}AL^{-\mathsf{T}}$ or with $B^{-1/2}AB^{-1/2}$. Methods based on the first two cases can be implemented with direct methods but this requires a factorization of B. Computing the Cholesky, or any other factorization of B is not always economically feasible for large problems. It is therefore important to explore alternatives based on the third case in which polynomial approximations of $B^{-1/2}$ are exploited.

All we need to apply the methods described in this paper is a way to compute $B^{-1/2}v$ or $B^{-1}v$ for an arbitrary vector v. These calculations amount to evaluating $f(B)v$ where $f(\lambda) = \lambda^{-1/2}$ in one case and $f(\lambda) = 1/\lambda$ in the other. Essentially the same method is used in both cases, in that $f(B)v$ is replaced by $f_k(B)v$ where f_k is an order k polynomial approximation to the function f obtained by a least-squares approach. Computing $B^{-1/2}v$, is a problem that was examined at length in the literature – see for example [2, 5, 14] and references therein. Here we use a simple scheme that relies on a Chebyshev approximation of the square root function in the interval $[a,\ b]$ where $a>0$.

Recall that any function that is analytic in $[a,\ b]$ can be expanded in Chebyshev polynomials. To do so, the first step is to map $[a,\ b]$ into the interval $[-1,\ 1]$, i.e., we impose the change of variables from $\lambda\in[a,\ b]$ to $t\in[-1,\ 1]$:

$$t = \frac{\lambda - c}{h} \quad \text{with} \quad c = \frac{a+b}{2}, \quad h = \frac{b-a}{2}\ .$$

In this way the function is transformed into a function f with variables in the interval $[-1,\ 1]$. It is this f that is approximated using the truncated Chebyshev expansion:

$$f_k(t) = \sum_{i=0}^{k} \gamma_i T_i(t) \quad \text{with} \quad \gamma_i = \frac{2-\delta_{i0}}{\pi} \int_{-1}^{1} \frac{f(s)T_i(s)}{\sqrt{1-s^2}} ds, \tag{2.4}$$

where $T_i(s)$ is the Chebyshev polynomial of the first kind of degree i. Here δ_{ij} is the Kronecker δ symbol so that $2-\delta_{k0}$ is equal to 1 when $k=0$ and to 2 otherwise.

Recall that T_i's are orthogonal with respect to the inner product

$$\langle p,q\rangle = \int_{-1}^{1} \frac{p(s)q(s)}{\sqrt{1-s^2}} ds. \tag{2.5}$$

We denote by $\|.\|_\infty$ the supremum norm and by $\|.\|_C$ the L_2 norm associated with the above dot product:

$$\|p\|_C = \left[\int_{-1}^{1} \frac{p(s)^2}{\sqrt{1-s^2}} ds\right]^{1/2}. \tag{2.6}$$

Note in passing that T_i's do not have a unit norm with respect to (2.6) but that the following normalized sequence is orthonormal:

$$\hat{T}_i(s) = \sqrt{\frac{2-\delta_{i0}}{\pi}} T_i(s), \tag{2.7}$$

so that (2.4) can be rewritten as $f_k(t) = \sum_{i=0}^{k} \hat{\gamma}_i \hat{T}_i(t)$ with $\hat{\gamma}_i = \langle f(t), \hat{T}_i(t)\rangle$.

The integrals in (2.4) are computed using Gauss-Chebychev quadrature. The accuracy of the approximation and therefore the degree needed to obtain a suitable approximation to use in replacement of $f(B)v$ depends essentially on the degree of smoothness of f. One issue here is to determine the number of integration points to use. Recall that when we use Gauss-Chebyshev quadrature with ν points, the calculated integral is exact for all polynomials of degree $\leq 2\nu-1$.

The reasoning for selecting ν is as follows. Let p_K be the truncated Chebyshev expansion of f, with $K \gg k$. Then for $i \leq k$ the coefficients $\hat{\gamma}_i$ for $i \leq k$ are the same for p_k and for p_K and they are:

$$\hat{\gamma}_i = \left\langle f, \hat{T}_i \right\rangle = \left\langle f - p_K, \hat{T}_i \right\rangle + \left\langle p_K, \hat{T}_i \right\rangle = \left\langle p_K, \hat{T}_i \right\rangle.$$

The last equality is due to the orthogonality of the error to the T_i's, when $i \leq K$. Now observe that since $p_K(t)\hat{T}_i(t)$ is a polynomial of degree $\leq K+k$ the integral $\langle p_K, \hat{T}_i\rangle$ will be computed exactly by the Gauss-Chebyshev rule as long as $K+k \leq 2\nu-1$, i.e., for $\nu \geq (K+k+1)/2$. For example, when $K=2k$ then for $\nu \geq (3k+1)/2$, $\hat{\gamma}_i$ *will be the exact coefficient not for* $f(t)$*, but for* p_{2k} *the degree* $2k$ *Chebyshev expansion which is usually much closer to* f *than* p_k. While $\nu = \lceil (3k+1)/2 \rceil$ is usually sufficient, we prefer a lower margin for error and select $\nu = 4k$ bearing in mind that the cost of quadrature is negligible.

2.2. Analysis of the approximation accuracy. Consider the two functions $f_1(\lambda) = \lambda^{-1/2}$ and $f_2(\lambda) = \lambda^{-1}$ over $\lambda \in [a,\ b]$ where $a > 0$. It is assumed that the interval $[a,\ b]$ contains the spectrum of B - with ideally $a = \lambda_{min}(B)$, $b = \lambda_{max}(B)$. We set $c = (a+b)/2$, $h = (b-a)/2$. As mentioned above we need to transform the interval $[a,\ b]$ into $[-1,\ 1]$, so the transformed functions being approximated are in fact

$$g(t) = (c+ht)^{-1/2}, \tag{2.8}$$

$$q(t) = (c+ht)^{-1}, \tag{2.9}$$

with the variable t now in $[-1,\ 1]$. These two functions are clearly analytic in the interval $[-1,\ 1]$ and they have a singularity when $c+ht=0$, i.e., at $t_s=-c/h$ which is less than -1. Existing results in the literature will help analyze the convergence of the truncated Chebyshev expansion in situations such as these, see, e.g., [31].

We can apply the result of Theorem 8.2 in the book [31] to show a strong convergence result. The Joukowsky transform $(z+1/z)/2$ maps the circle $C(0,\rho)$ into an ellipse E_ρ, with major semi-axis $(\rho+\rho^{-1})/2$ and focii $-1, 1$. There are two values of ρ that give the same ellipse and they are inverses of each other. We assume that $\rho>1$. The ellipse E_ρ is called the Bernstein ellipse in the framework of the theorem in [31] which is restated below for the present context. See Fig. 2.1 for an illustration of Bernstein ellipses corresponding to different ρ's.

THEOREM 2.1. *[31, Theorem 8.2] Let a function f analytic in $[-1,\ 1]$ be analytically continuable to the open Bernstein ellipse E_ρ where it satisfies $|f(t)|\le M(\rho)$ for some $M(\rho)$. Then for each $k\ge 0$, its truncated Chebyshev expansion f_k (eq. (2.4)) satisfies:*

$$\|f-f_k\|_\infty \le \frac{2M(\rho)\rho^{-k}}{\rho-1}. \tag{2.10}$$

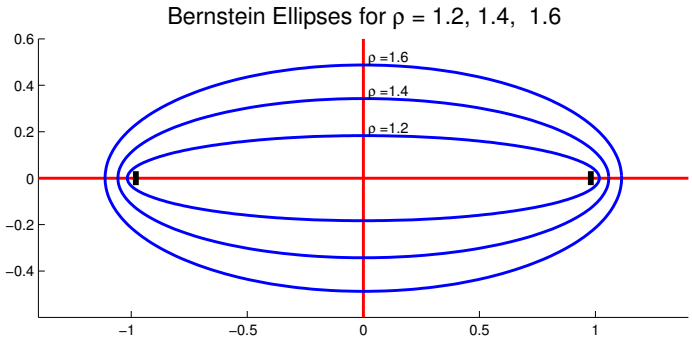


FIG. 2.1. *Bernstein ellipses for $\rho=1.2, 1.4, 1.6$.*

The Bernstein ellipse should not contain the point of singularity. Therefore, for the two functions under consideration, we should take *any* $\rho>1$ such that $(\rho+\rho^{-1})/2<c/h$, i.e., ρ must satisfy:

$$1<\rho<\frac{c}{h}+\sqrt{\left(\frac{c}{h}\right)^2-1}. \tag{2.11}$$

The next ingredient from the theorem is an upper bound $M(\rho)$ for $|f(t)|$ in E_ρ. In fact the maximum value of this modulus is computable for both functions under consideration and it is given in the next lemma.

LEMMA 2.2. *Let ρ be given such that (2.11) is satisfied. Then the maximum modulii of the functions (2.8) and (2.9) for $t\in E_\rho$ are given, respectively, by*

$$M_g(\rho)=\frac{1}{\sqrt{c-h\frac{\rho+\rho^{-1}}{2}}}, \tag{2.12}$$

$$M_q(\rho)=\frac{1}{c-h\frac{\rho+\rho^{-1}}{2}}. \tag{2.13}$$

Proof. Denote $d = c + ht$ the term inside the parentheses of (2.8) and (2.9) and write $t \in E_\rho$ as: $t = \frac{1}{2}[\rho e^{i\theta} + \rho^{-1} e^{-i\theta}]$. Then $d = c + h(\rho e^{i\theta} + \rho^{-1} e^{-i\theta})/2$ and

$$\begin{aligned} |d|^2 &= (c + ht)(c + h\bar{t}) = c^2 + hc(t + \bar{t}) + h^2 t\bar{t} \\ &= c^2 + hc(\rho + \rho^{-1}) \cos\theta + \frac{h^2}{4}[\rho^2 + \rho^{-2} + 2\cos(2\theta)]. \end{aligned}$$

Observe that $\rho^2 + \rho^{-2} = (\rho + \rho^{-1})^2 - 2$ and $\cos(2\theta) = 2\cos^2\theta - 1$. Therefore,

$$\begin{aligned} |d|^2 &= c^2 + hc(\rho + \rho^{-1}) \cos\theta + \frac{h^2}{4}[(\rho + \rho^{-1})^2 - 2(1 - \cos(2\theta))] \\ &= c^2 + hc(\rho + \rho^{-1}) \cos\theta + \frac{h^2}{4}[(\rho + \rho^{-1})^2 - 4(1 - \cos^2\theta)] \\ &= \left[c + h\frac{\rho + \rho^{-1}}{2} \cos\theta\right]^2 + \frac{h^2}{4}\left[(\rho + \rho^{-1})^2 - 4\right](1 - \cos^2\theta) \\ &= \left[c + h\frac{\rho + \rho^{-1}}{2} \cos\theta\right]^2 + h^2\left[\left(\frac{\rho + \rho^{-1}}{2}\right)^2 - 1\right] \sin^2\theta. \end{aligned}$$

Since $(\rho + \rho^{-1})/2 > 1$, the second term in brackets is positive and it is then clear that the minimum value of $|d|^2$ is reached when $\theta = \pi$ and the corresponding $|d|$ is $c - h(\rho + \rho^{-1})/2$. Inverting this gives (2.13). Taking the inverse square root yields (2.12) and this completes the proof. □

Note that, as expected, both maxima go to infinity as ρ approaches its right (upper) bound given by (2.11). We can now state the following theorem which simply applies Theorem 2.1 to the functions (2.8) and (2.9), using the bounds for $M(\rho)$ obtained in Lemma 2.2.

THEOREM 2.3. *Let g and q be the functions given by (2.8) and (2.9) and let ρ be any real number that satisfies the inequalities (2.11). Then the truncated Chebyshev expansions g_{k_1} and q_{k_2} of g and q, respectively, satisfy:*

$$\|g - g_{k_1}\|_\infty \leq \frac{2\rho^{-k_1}}{(\rho - 1)\sqrt{c - h\frac{\rho + \rho^{-1}}{2}}}, \tag{2.14}$$

$$\|q - q_{k_2}\|_\infty \leq \frac{2\rho^{-k_2}}{(\rho - 1)\left(c - h\frac{\rho + \rho^{-1}}{2}\right)}. \tag{2.15}$$

Theorem 8.1 in [31], upon which Theorem 2.1 is based, states that the coefficients γ_i in (2.4) decay geometrically, i.e.,

$$|\gamma_k| \leq 2M(\rho)\rho^{-k}. \tag{2.16}$$

Based on the above inequality, it is now possible to establish the following result for the approximation error of g_{k_1} and q_{k_2} measured in the Chebyshev norm.

PROPOSITION 2.4. *Under the same assumptions as for Theorem 2.3, the trun-*

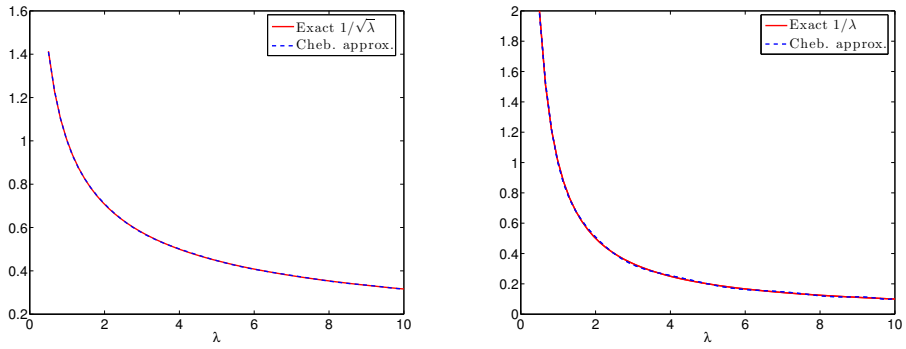


FIG. 2.2. *Degree* 10 *Chebyshev polynomial approximations to* $1/\sqrt{\lambda}$ *and* $1/\lambda$ *on the interval* [0.5, 10].

cated Chebyshev expansions g_{k_1} *and* q_{k_2} *of* g *and* q*, satisfy, respectively:*

$$\|g - g_{k_1}\|_C \leq \sqrt{\frac{2\pi}{\rho^2 - 1}} \frac{\rho^{-k_1}}{\sqrt{c - h\frac{\rho+\rho^{-1}}{2}}}, \tag{2.17}$$

$$\|q - q_{k_2}\|_C \leq \sqrt{\frac{2\pi}{\rho^2 - 1}} \frac{\rho^{-k_2}}{\left(c - h\frac{\rho+\rho^{-1}}{2}\right)}. \tag{2.18}$$

Proof. For any function f expandable as in (2.4), we have

$$f(t) - f_k(t) = \sum_{i=k+1}^{\infty} \gamma_i T_i(t).$$

Because of the orthogonality of the Chebyshev polynomials and the inequality (2.16), we obtain

$$\begin{aligned}\|f(t) - f_k(t)\|_C^2 = \sum_{i=k+1}^{\infty} |\gamma_i|^2 \|T_i(t)\|_C^2 &\leq \sum_{i=k+1}^{\infty} 4M(\rho)^2 \rho^{-2i} \frac{\pi}{2} \\ &\leq 2M(\rho)^2 \pi \rho^{-2(k+1)} \frac{1}{1-\rho^{-2}} = 2M(\rho)^2 \pi \rho^{-2k} \frac{1}{\rho^2 - 1}.\end{aligned}$$

Taking the square root and replacing the values of $M(\rho)$ from Lemma 2.2 yield the two inequalities. ∎

Both Theorem 2.3 and Proposition 2.4 show that the Chebyshev expansions g_{k_1} and q_{k_2} converge geometrically. The plot in Fig. 2.2 indicates that a low degree is sufficient to reach a reasonable accuracy for the needs of computing the DOS.

2.3. Bounds involving the condition number of B**.** Theorem 2.3 shows that the *asymptotic* convergence rate increases with ρ. However, the "optimal" value of ρ, i.e., the one that yields the smallest bounds in (2.14) or (2.15), depends on k_i and is hard to choose in practice. Here, we will discuss two simple choices for ρ that will help analyze the convergence. First, we select $\rho = \rho_0 \equiv c/h$ which satisfies the bounds (2.11). It leads to

$$M_g(\rho_0) = \frac{\sqrt{2}}{\sqrt{c - h^2/c}}, \qquad M_q(\rho_0) = (M_g(\rho_0))^2 \ . \tag{2.19}$$

Note that in the context of our problem, if we denote by $\lambda_{max}(B), \lambda_{min}(B)$ the largest and smallest eigenvalues of B and by κ its spectral condition number, then

$$\rho_0 = c/h = \frac{\lambda_{max}(B) + \lambda_{min}(B)}{\lambda_{max}(B) - \lambda_{min}(B)} = \frac{\kappa + 1}{\kappa - 1},$$

and therefore, for this choice of ρ, the bounds of the theorem evolve asymptotically like $(\frac{\kappa-1}{\kappa+1})^k$. A slightly more elaborate selection of ρ is the value for which $(\rho + \rho^{-1})/2 = \sqrt{c/h}$ which is $\rho_1 = \sqrt{c/h} + \sqrt{(c/h) - 1}$. For $t \geq 1$, $t + \sqrt{t^2 - 1}$ is an increasing function and therefore, $1 \leq \rho_1 \leq (c/h) + \sqrt{(c/h)^2 - 1}$ and so the bounds (2.11) are satisfied. With this we get:

$$M_g(\rho_1) = \frac{1}{\sqrt{c - \sqrt{hc}}}, \qquad M_q(\rho_1) = (M_g(\rho_1))^2 \ .$$

In addition, we note that ρ_1 can also be expressed in terms of the spectral condition number κ of B as follows: $\rho_1 = [\sqrt{\kappa + 1} + \sqrt{2}]/[\sqrt{\kappa - 1}]$. The resulting term ρ_1^{-k} in (2.14) and (2.15) will decay much faster than ρ_0^{-k} when κ is larger than 2. Both choices of ρ show that for a fixed degree k, a smaller κ will result in faster convergence.

If B is a mass matrix obtained from a FEM discretization, κ can become very large for a general nonuniform mesh. One simple technique to reduce the value of κ is to use diagonal scaling [17, 35, 36]. Suppose $D = \mathrm{diag}(B)$, then by congruence, the following problem has the same eigenvalues as (1.3)

$$D^{-1/2} A D^{-1/2} z = \lambda D^{-1/2} B D^{-1/2} z, \quad \text{with} \quad z = D^{1/2} x. \tag{2.20}$$

It was shown in [35, 36] that, for any conforming mesh of tetrahedral (P1) elements in three dimensions, $\kappa(D^{-1/2} B D^{-1/2})$ is bounded by 5 and for a mesh of rectangular bilinear (Q1) elements in two dimensions, $\kappa(D^{-1/2} B D^{-1/2})$ is bounded by 9. Moreover, this diagonal scaling technique has also been exploited to reduce the spectral condition number of graph Laplacians in the network analysis [3]. As a result, we will always preprocess the matrix pencil (A, B) by diagonal scaling before computing the DOS.

With the approximations in (2.4), we obtain

$$B^{-1} \approx g_{k_1}(B) := \sum_{i=0}^{k_1} \gamma_i T_i[(B - cI)/h], \tag{2.21}$$

$$B^{-1/2} \approx q_{k_2}(B) := \sum_{i=0}^{k_2} \beta_i T_i[(B - cI)/h]. \tag{2.22}$$

Using the above approximations to replace B^{-1} and $B^{-1/2}$ in (2.1) and (2.3), will amount to computing the DOS of the modified problem

$$g_{k_1}(B) A \tilde{x} = \tilde{\lambda} \tilde{x}. \tag{2.23}$$

Therefore, it is important to show that the distance between $\tilde{\lambda}$ and λ is small when g_{k_1} and q_{k_2} reach a certain accuracy. We will need the following perturbation result for Hermitian definite pencils.

THEOREM 2.5. *[21, Theorem 2.2] Suppose that a Hermitian definite pencil (A, B) has eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. If $\Delta A, \Delta B$ are Hermitian and $||\Delta B||_2 <$*

$\lambda_{min}(B)$, then $(A+\Delta A, B+\Delta B)$ is a Hermitian definite pencil whose eigenvalues $\hat{\lambda}_1 \le \hat{\lambda}_2 \le \cdots \le \hat{\lambda}_n$ satisfy

$$|\lambda_i - \hat{\lambda}_i| \le \frac{||\Delta A||_2}{\lambda_{min}(B)} + \frac{|\lambda_i|\lambda_{min}(B) + ||\Delta A||_2}{\lambda_{min}(B)(\lambda_{min}(B) - ||\Delta B||_2)}||\Delta B||_2. \tag{2.24}$$

In the context of (2.23), the perturbation ΔB in Theorem 2.6 corresponds to the approximation error of $g_{k_1}(B)$ to B^{-1}. This implies that we can rewrite (2.23) in the form of

$$A\tilde{x} = \tilde{\lambda}(B + \Delta B)\tilde{x} \quad \text{with} \quad \Delta B = (g_{k_1}(B))^{-1} - B,$$

and then apply Theorem 2.5 to prove the following perturbation bound for (2.23).

THEOREM 2.6. *Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of $B^{-1}A$ and $\hat{\lambda}_1 \le \hat{\lambda}_2 \le \cdots \le \hat{\lambda}_n$ be the eigenvalues of $g_{k_1}(B)A$. If $||g - g_{k_1}||_\infty \le \tau$ and $||B||_2 \le 1/\tau$, then we have*

$$|\lambda_i - \hat{\lambda}_i| \le \frac{|\lambda_i|}{\lambda_{min}(B) - ||\Delta B||_2}||\Delta B||_2, \tag{2.25}$$

with $||\Delta B||_2 \le \frac{||B||_2^2}{1-||B||_2\tau}\tau$.

Proof. Denote by $\theta_1, \theta_2, \ldots, \theta_n$ the eigenvalues of B^{-1} and $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n$ the eigenvalues of $g_{k_1}(B)$. Since $||g - g_{k_1}||_\infty \le \tau$, we have

$$||B^{-1} - g_{k_1}(B)||_2 = \max_i |\theta_i - \hat{\theta}_i| \le \tau. \tag{2.26}$$

On the other hand, we know that

$$||\Delta B||_2 = ||B - (g_{k_1}(B))^{-1}||_2 = \max_i |1/\theta_i - 1/\hat{\theta}_i| \le \frac{\tau}{\theta_1(\theta_1 - \tau)} = \frac{||B||_2^2}{1 - ||B||_2\tau}\tau. \tag{2.27}$$

Substituting $||\Delta A||_2$ and $||\Delta B||_2$ in (2.24) with 0 and (2.27), respectively, we obtain the bound (2.25). ∎

Theorem 2.6 indicates that if the degree of the Chebyshev expansions is chosen in such a way that the bounds (2.12–2.13) are less than or equal to τ, the eigenvalues of (2.23) would be close enough to those of (2.1). In the next two sections, we will show how to extend the standard algorithms for computing the DOS to generalized eigenvalue problems of the form (2.23).

3. The Kernel Polynomial Method. The Kernel Polynomial Method (KPM) is an effective technique proposed by physicists and chemists in the mid-1990s [7, 22, 27, 28, 29, 34] to calculate the DOS of a Hermitian matrix A. Its essence is to expand the function ϕ in (1.1), which is a sum of Dirac δ-functions, into Chebyshev polynomials.

3.1. Background: The KPM for standard eigenvalue problems. As is the case for all methods which rely on Chebyshev expansions, a change of variables is first performed to map the interval $[\lambda_{\min},\ \lambda_{\max}]$ into $[-1,\ 1]$. We assume this is already performed and so the eigenvalues are in the interval $[-1,\ 1]$. To estimate the spectral density function (1.1), the KPM method approximates $\phi(t)$ by a finite expansion in a basis of orthogonal polynomials, in this case, Chebyshev polynomials

of the first kind. Following the Silver-Röder paper [27], we include, for convenience, the inverse of the weight function into the spectral density function, so we expand instead the distribution:

$$\hat{\phi}(t) = \sqrt{1-t^2}\phi(t) = \sqrt{1-t^2} \times \frac{1}{n}\sum_{j=1}^{n}\delta(t-\lambda_j). \tag{3.1}$$

Then, we have the (full) expansion

$$\hat{\phi}(t) = \sum_{k=0}^{\infty}\mu_k T_k(t), \tag{3.2}$$

where the expansion coefficients μ_k are formally defined by

$$\mu_k = \frac{2-\delta_{k0}}{\pi}\int_{-1}^{1}\frac{1}{\sqrt{1-t^2}}T_k(t)\hat{\phi}(t)dt = \frac{2-\delta_{k0}}{n\pi}\sum_{j=1}^{n}T_k(\lambda_j).$$

Thus, apart from the scaling factor $(2-\delta_{k0})/(n\pi)$, μ_k is the trace of $T_k(A)$ and this can be estimated by various methods including, but not limited to, stochastic approaches. There are variations on this idea starting with the use of different orthogonal polynomials, to alternative ways in which the traces can be estimated.

The standard stochastic argument for estimating $\mathrm{Trace}(T_k(A))$, see [15, 27, 30], entails generating a large number of random vectors $v_0^{(1)}, v_0^{(2)}, \cdots, v_0^{(n_{\mathrm{vec}})}$ with each component obtained from a normal distribution with zero mean and unit standard deviation, and each vector is normalized such that $\|v_0^{(l)}\|_2 = 1, l = 1, \ldots, n_{\mathrm{vec}}$. The subscript 0 is added to indicate that the vector has not been multiplied by the matrix A. Then we can estimate the trace of $T_k(A)$ as follows:

$$\mathrm{Trace}(T_k(A)) \approx \frac{1}{n_{\mathrm{vec}}}\sum_{l=1}^{n_{\mathrm{vec}}}\left(v_0^{(l)}\right)^T T_k(A)v_0^{(l)}, \tag{3.3}$$

where the error decays as $\frac{1}{\sqrt{n_{nev}}}$ [15]. Then this will lead to the desired estimate:

$$\mu_k \approx \frac{2-\delta_{k0}}{n\pi n_{\mathrm{vec}}}\sum_{l=1}^{n_{\mathrm{vec}}}\left(v_0^{(l)}\right)^T T_k(A)v_0^{(l)}. \tag{3.4}$$

Consider the computation of each term $v_0^T T_k(A)v_0$ (the superscript l is dropped for simplicity). The 3-term recurrence of the Chebyshev polynomial: $T_{k+1}(t) = 2tT_k(t) - T_{k-1}(t)$ can be exploited to compute $T_k(A)v_0$, so that, if we let $v_k \equiv T_k(A)v_0$, we have

$$v_{k+1} = 2Av_k - v_{k-1}. \tag{3.5}$$

The approximate density of states will be limited to Chebyshev polynomials of degree m, so ϕ is approximated by the truncated expansion:

$$\widetilde{\phi}_m(t) = \frac{1}{\sqrt{1-t^2}}\sum_{k=0}^{m}\mu_k T_k(t). \tag{3.6}$$

It has been proved in [19] that the expansion error in (3.6) decays as ρ^{-m} for some constant $\rho > 1$.

For a general matrix A whose eigenvalues are not necessarily in the interval $[-1,\ 1]$, a linear transformation is first applied to A to bring its eigenvalues to the desired interval. Specifically, we will apply the method to the matrix

$$\tilde{A} = \frac{A - cI}{h}, \tag{3.7}$$

where

$$c = \frac{\lambda_{\min} + \lambda_{\max}}{2}\ , \quad h = \frac{\lambda_{\max} - \lambda_{\min}}{2}. \tag{3.8}$$

It is important to ensure that the eigenvalues of $\tilde{A}$ are within the interval $[-1,\ 1]$. In an application requiring a similar approach [38], we obtain the upper and lower bounds of the spectrum from Ritz values provided by a standard Lanczos iteration. We ran m Lanczos steps but extended the interval $[\lambda_{\min},\ \lambda_{\max}]$ by using the bounds obtained from the Lanczos algorithm. Specifically, the upper bound is set to $\widetilde{\lambda}_m + \eta$ where $\eta = \|(A - \widetilde{\lambda}_m I)\widetilde{u}_m\|_2$, and $(\widetilde{\lambda}_m, \widetilde{u}_m)$ is the (algebraically) largest Ritz pair of A. In a similar way, the lower bound is set to $\widetilde{\lambda}_1 - \beta$ where $\beta = \|(A - \widetilde{\lambda}_1 I)\widetilde{u}_1\|_2$ and $(\widetilde{\lambda}_1, \widetilde{u}_1)$ is the (algebraically) smallest Ritz pair of A. To summarize, we outline the major steps of the KPM for approximating the spectral density of a Hermitian matrix in Algorithm 1.

Algorithm 1 The Kernel Polynomial Method

Input: A Hermitian matrix A, a set of points $\{t_i\}$ at which DOS is to be evaluated, the degree m of the expansion polynomial
Output: Approximate DOS evaluated at $\{t_i\}$

1: Compute the upper bound and the lower bound of the spectrum of A
2: Compute c and h in (3.8) with those bounds
3: Set $\mu_k = 0$ for $k = 0, \cdots, m$
4: **for** $l = 1 : n_{\text{vec}}$ **do**
5: Select a new random vector $v_0^{(l)}$
6: **for** $k = 0 : m$ **do**
7: Compute $T_k((A - cI)/h)v_0^{(l)}$ using 3-term recurrence (3.5)
8: Update μ_k using (3.4)
9: **end for**
10: **end for**
11: Evaluate the average value of $\{\widetilde{\phi}_m((t_i - c)/h)\}$ at the given set of points $\{t_i\}$ using (3.6)

3.2. The KPM for generalized eigenvalue problems. We now return to the generalized problem (1.3). Generalizing the KPM algorithm to this case is straightforward when the square root factorization $B = S^2$ or the Cholesky factorization $B = LL^T$ is available: we just need to use Algorithm 1 with A replaced by $S^{-1}AS^{-1}$ or $L^{-1}AL^{-T}$. In this section we only discuss the case where a square root factorization is used. The alternative of using the Cholesky factorization can be carried out in a similar way. Clearly $S^{-1}AS^{-1}$ needs not be explicitly computed. Instead, the product $S^{-1}AS^{-1}w$ that is required when computing $T_k((S^{-1}AS^{-1} - cI)/h)v_0^{(l)}$ in Line 7 of Algorithm 1, can be approximated by matrix-vector products with $q_{k_2}(B)$ in (2.22) and the matrix A.

The important point here is that if we simply follow the 3-term recurrence (3.5) and let $v_k \equiv T_k((S^{-1}AS^{-1} - cI)/h)v_0$, we have

$$v_{k+1} = 2\frac{S^{-1}AS^{-1} - cI}{h}v_k - v_{k-1}. \tag{3.9}$$

This implies that the computation of each v_k will involve two matrix-vector products with S^{-1} and one matrix-vector product with A. On the other hand, premultiplying both sides of (3.9) with S^{-1} leads to

$$\begin{aligned} S^{-1}v_{k+1} &= 2S^{-1}\frac{S^{-1}AS^{-1} - cI}{h}v_k - S^{-1}v_{k-1} \\ &= 2\frac{B^{-1}A - cI}{h}S^{-1}v_k - S^{-1}v_{k-1}. \end{aligned}$$

Denoting by $w_k := S^{-1}v_k$, we obtain another 3-term recurrence

$$w_{k+1} = 2\frac{B^{-1}A - cI}{h}w_k - w_{k-1} \quad \text{with} \quad w_0 = S^{-1}v_0. \tag{3.10}$$

Now the computation of each w_k only involves one matrix-vector product with B^{-1} and one matrix-vector product with A. Since Theorem 2.3 shows that both the approximation errors of g_{k_1} and q_{k_2} decay as ρ^{-k_i}, this indicates that the same approximation accuracy will likely lead to roughly the same degree for g_{k_1} and q_{k_2}. As a result, recurrence (3.10) is computationally more economical than recurrence (3.9) when we replace B^{-1} and S^{-1} with $g_{k_1}(B)$ in (2.21) and $q_{k_2}(B)$ in (2.22), respectively. In the end, $v_0^T T_k((S^{-1}AS^{-1} - cI)/h)v_0$ in (3.4) is computed as $w_0^T B w_k$.

Similarly, if Cholesky factorization of B is applied, then the following 3-term recurrence is preferred in actual computations

$$w_{k+1} = 2\frac{B^{-1}A - cI}{h}w_k - w_{k-1} \quad \text{with} \quad w_0 = L^{-T}v_0. \tag{3.11}$$

4. The Lanczos method for Density of States. The well-known connection between the Gaussian Quadrature and the Lanczos algorithm has also been exploited to compute the DOS [20]. We first review the method for standard problems before extending it to matrix pencils.

4.1. Background: The Lanczos procedure for the standard DOS. The Lanczos algorithm builds an orthonormal basis $V_m = [v_1, v_2, \cdots, v_m]$ for the *Krylov subspace:* $\text{Span}\{v_1, Av_1, \cdots, A^{m-1}v_1\}$ with an initial vector v_1. See Algorithm 4.1 for a summary.

Algorithm 2 Lanczos algorithm for a Hermitian matrix A

1: Choose an initial vector v_1 with $\|v_1\|_2 = 1$ and set $\beta_1 = 0$, $v_0 = 0$
2: **for** $j = 1, 2, \ldots, m$ **do**
3: $\quad w := Av_j - \beta_j v_{j-1}$
4: $\quad \alpha_j = (w, v_j)$
5: $\quad w := w - \alpha_j v_j$
6: $\quad$ Full reorthogonalization: $w := w - \sum_i (w, v_i)v_i$ for $i \le j$
7: $\quad \beta_{j+1} = \|w\|_2$
8: $\quad$ If $\beta_{j+1} == 0$ restart or exit
9: $\quad v_{j+1} := w/\beta_{j+1}$
10: **end for**

At the completion of m steps of Algorithm 4.1, we end up with the factorization $V_m^T A V_m = T_m$ - with

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \beta_4 & & \\ & & . & . & . & \\ & & & . & . & \beta_m \\ & & & & \beta_m & \alpha_m \end{pmatrix}.$$

Note that the vectors v_j, for $j = 1, \cdots, m$, satisfy the 3-term recurrence

$$\beta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1}.$$

In theory the v_j's defined by this recurrence are orthonormal. In practice there is a severe loss of orthogonality and a form of reorthogonalization (Line 6 in Algorithm 2) is necessary.

Let $\theta_i,\ i = 1 \cdots, m$ be the eigenvalues of T_m. These are termed *Ritz values*. If $\{y_i\}_{i=1:m}$, are the associated eigenvectors, then the vectors $\{V_m y_i\}_{i=1:m}$ are termed *Ritz vectors* and they represent corresponding approximate eigenvectors of A. Typically, eigenvalues of A on both ends of the spectrum are first well approximated by corresponding eigenvalues of T_m (Ritz values) and, as more steps are taken, more and more eigenvalues toward the inside of the spectrum become better approximations. Thus, one can say that the Ritz values approximate the eigenvalues of A progressively from 'outside in'.

One approach to compute the DOS is to compute these θ_i's and then get approximate DOS from them. However, the θ_i's tend to provide poor approximations to the eigenvalues located at the interior of the spectrum and so this approach does not work too well in practice. A better idea is to exploit the relation between the Lanczos procedure and the (discrete) orthogonal polynomials and the related Gaussian quadrature.

Assume the initial vector v_1 in the Lanczos method can be expanded in the eigenbasis of A as $v_1 = \sum_{i=1}^{n} \omega_i u_i$. Then the Lanczos process builds orthogonal polynomials with respect to the discrete (Stieljes) inner product:

$$\int_a^b f(t) q(t) d\mu(t) \equiv (f(A)v_1, q(A)v_1), \tag{4.1}$$

where the measure $\mu(t)$ is a piecewise constant function defined as

$$\mu(t) = \begin{cases} 0, & \text{if } t < a = \lambda_1, \\ \sum_{j=1}^{i-1} \omega_j^2, & \text{if } \lambda_{i-1} \le t < \lambda_i,\ i = 2:n, \\ \sum_{j=1}^{n} \omega_j^2, & \text{if } b = \lambda_n \le t. \end{cases} \tag{4.2}$$

In particular, when $q(t) = 1$, (4.1) takes the form of

$$\int_a^b f(t) d\mu(t) \equiv (f(A)v_1, v_1), \tag{4.3}$$

which we will refer to as the Stieljes integral of f. Golub and Welsh [12] showed how to extract Gaussian-quadrature formulas for integrals of the type shown above. The

integration nodes for a Gaussian quadrature formula with m points, are simply the eigenvalue values $\theta_i, i = 1, \cdots, m$ of T_m. The associated weights are the squares of the first components of the eigenvectors associated with θ_i's. Thus,

$$\int_a^b f(t)d\mu(t) \approx \sum_{i=1}^{m} a_i f(\theta_i) \ , \quad a_i = \left[e_1^T y_i\right]^2 . \tag{4.4}$$

As is known, such an integration formula is exact for polynomials of degree up to $2m-1$, see, e.g., [11, 12]. Then we will derive an approximation to the DOS with the quadrature rule (4.4).

The Stieljes integral $\int_a^b f(t)d\mu(t)$ satisfies the following equality:

$$\int_a^b f(t)d\mu(t) = (f(A)v_1, v_1) = \sum_{i=1}^{n} \omega_i^2 f(\lambda_i).$$

We can view this as a distribution ϕ_{v_1} applied to f:

$$(f(A)v_1, v_1) \equiv \langle \phi_{v_1}, f \rangle \quad \text{with} \quad \phi_{v_1} \equiv \sum_{i=1}^{n} \omega_i^2 \delta(t - \lambda_i). \tag{4.5}$$

Assume for a moment that we are able to find a special vector v_1 which satisfies $\omega_i^2 = 1/n$ for all i. Then the above distribution becomes $\phi_{v_1} = \frac{1}{n}\sum_{i=1}^{n} \delta(t-\lambda_i)$ which is exactly the DOS defined in (1.2). Next, we consider how ϕ_{v_1} can be approximated via Gaussian-quadrature. Based on (4.4) and (4.5), we know that

$$\langle \phi_{v_1}, f \rangle \equiv (f(A)v_1, v_1) = \int_a^b f(t)d\mu(t) \approx \sum_{i=1}^{m} a_i f(\theta_i) \equiv \left\langle \sum_{i=1}^{m} a_i \delta(t-\theta_i), f \right\rangle .$$

Since f is an arbitrary function, we obtain the following approximation expressed for the DOS:

$$\phi_{v_1} \approx \tilde{\phi}_{v_1} := \sum_{i=1}^{m} a_i \delta(t - \theta_i). \tag{4.6}$$

In the next theorem, we show that the approximation error of the Lanczos method for computing the DOS decays as ρ^{-2m} for a constant $\rho > 1$. Here, we follow (2.5) in [20] to measure the approximation error between ϕ and $\tilde{\phi}_{v_1}$ as

$$|\langle \phi, g \rangle - \left\langle \tilde{\phi}_{v_1}, g \right\rangle|, \quad \text{with } g(t) \text{ being an analytic function on } [-1, 1].$$

THEOREM 4.1. *Assume $A \in \mathbb{C}^{n\times n}$ is a Hermitian matrix with its spectrum inside $[-1,\ 1]$. If $v_1 \in \mathbb{R}^n$ is a unit vector with equal weights in all eigenvectors of A, then the approximation error of a m-term expansion (4.6) is*

$$|\langle \phi, g \rangle - \left\langle \tilde{\phi}_{v_1}, g \right\rangle| \leq \frac{4\rho^2 M(\rho)}{(\rho^2 - 1)\rho^{2m}}, \tag{4.7}$$

where $\rho > 1$ and $M(\rho)$ are constants.

Proof. Let p_{2m-1} be the Chebyshev polynomial approximation of degree $2m-1$ to $g(t)$:

$$p_{2m-1}(t) = \sum_{k=0}^{2m-1} \gamma_k T_k(t) \approx g(t) = \sum_{k=0}^{\infty} \gamma_k T_k(t).$$

Since the quadrature formula (4.4) is exact for polynomials with degree up to $2m-1$, we have

$$\int_a^b p_{2m-1}(t)d\mu(t) = \sum_{i=1}^{m} a_i p_{2m-1}(\theta_i).$$

Therefore, we get

$$\begin{aligned}
|\langle \phi, g\rangle - \left\langle \tilde{\phi}_{v_1}, g\right\rangle| &= |\frac{1}{n}\sum_{i=1}^{n} g(\lambda_i) - \sum_{j=1}^{m} a_j g(\theta_j)| = |\int_a^b g(t)d\mu(t) - \sum_{j=1}^{m} a_j g(\theta_j)| \\
&\le |\int_a^b g(t) - p_{2m-1}(t)d\mu(t)| + |\int_a^b p_{2m-1}(t)d\mu(t) - \sum_{j=1}^{m} a_j g(\theta_j)| \\
&\le \int_a^b |g(t) - p_{2m-1}(t)|d\mu(t) + \sum_{j=1}^{m} a_j |p_{2m-1}(\theta_j) - g(\theta_j)| \\
&\le \sum_{k=2m}^{\infty} \int_a^b |\gamma_k||T_k(t)|d\mu(t) + \sum_{j=1}^{m} a_j \sum_{k=2m}^{\infty} |\gamma_k||T_k(\theta_j)|.
\end{aligned}$$

Based on (2.16), we know that

$$\sum_{k=2m}^{\infty} |\gamma_k||T_k(\theta_j)| \le \sum_{k=2m}^{\infty} 2M(\rho)\rho^{-k}|T_k(\theta_j)| \le \sum_{k=2m}^{\infty} 2M(\rho)\rho^{-k}.$$

Since $\sum_{j=1}^{m} a_j = \int_a^b d\mu(t) = (v_1, v_1) = 1$, we have

$$\sum_{j=1}^{m} a_j \sum_{k=2m}^{\infty} |\gamma_k||T_k(\theta_j)| \le \frac{2\rho^2 M(\rho)}{(\rho^2-1)\rho^{2m}} \sum_{j=1}^{m} a_j = \frac{2\rho^2 M(\rho)}{(\rho^2-1)\rho^{2m}}. \tag{4.8}$$

For the first term, we have

$$\int_a^b |T_k(t)|d\mu(t) = \frac{1}{n}\sum_j |T_k(\lambda_j)| \le 1,$$

and therefore,

$$\sum_{k=2m}^{\infty} \int_a^b |\gamma_k||T_k(t)|d\mu(t) \le \sum_{k=2m}^{\infty} 2M(\rho)\rho^{-k} \le \frac{2\rho^2 M(\rho)}{(\rho^2-1)\rho^{2m}}. \tag{4.9}$$

Adding the bounds in (4.8) and (4.9), we obtain (4.7). ∎

Theorem 4.1 indicates that the approximation error from the Lanczos method for computing the DOS decays as ρ^{-2m}, which is twice as fast as the KPM method with degree m.

The approximation in (4.6) is achieved by taking an idealistic vector v_1 that has equal weights $(\pm 1/\sqrt{n})$ in all eigenvectors in its representation in the eigenbasis. A common strategy to mimic the effect of having a vector with $\mu_i = 1/\sqrt{n}, \forall i$, is to use s random vectors $v_1^{(k)}$, called sample vectors, and average the results of the above formula over them:

$$\phi \approx \frac{1}{s}\sum_{k=1}^{s}\sum_{i=1}^{m} a_i^{(k)}\delta(t-\theta_i^{(k)}). \tag{4.10}$$

Here the superscript (k) relates to the k-th sample vector and $\theta_i^{(k)}$, $a_i^{(k)}$ are the nodes and weights of the quadrature formula shown in (4.4) for this sample vector.

4.2. Generalized problems. A straightforward way to deal with the generalized case is to apply the standard Lanczos algorithm (Algorithm 2) described in the previous section to the matrix $S^{-1}AS^{-1}$ (or $L^{-1}AL^{-T}$). This leads to the relation:

$$S^{-1}AS^{-1}V_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^{\mathsf{T}}. \tag{4.11}$$

If we set $W_m = S^{-1}V_m$, and multiply through by S^{-1}, then we get

$$B^{-1}AW_m = W_m T_m + \beta_{m+1} w_{m+1} e_m^{\mathsf{T}}, \tag{4.12}$$

where it is important to note that W_m is B-orthogonal since

$$W_m^{\mathsf{T}} B W_m = V_m^{\mathsf{T}} S^{-1} B S^{-1} V_m = V_m^{\mathsf{T}} V_m = I.$$

It is possible to generate a basis V_m of the Krylov subspace $K_m(v_1, S^{-1}AS^{-1})$ if we want to deal with the standard problem with $S^{-1}AS^{-1}$. It is also possible to generate the basis W_m of the Krylov subspace $K_m(w_1, B^{-1}A)$ directly if we want to deal with the standard problem with $B^{-1}A$ using the B-inner product. From our discussion at the end of Section 3.2, we know that the second case is computationally more efficient.

Now let us focus on the case (4.12). If we start the Lanczos algorithm with a vector w_1 where $\|w_1\|_B = 1$, we could generate the sequence w_i through Algorithm 3, which is described as Algorithm 9.2 in [23, p.230].

Algorithm 3 Lanczos algorithm for matrix pair (A, B)

1: Choose an initial vector w_1 with $\|w_1\|_B = 1$. Set $\beta_1 = 0$, $w_0 = 0$, $z_0 = 0$, and compute $z_1 = Bw_1$
2: **for** $j = 1, 2, \ldots, m$ **do**
3: $z := Aw_j - \beta_j z_{j-1}$
4: $\alpha_j = (z, w_j)$
5: $z := z - \alpha_j z_j$
6: Full reorthogonalization: $z := z - \sum_i (z, w_i) z_i$ for $i \leq j$
7: $w := B^{-1}z$
8: $\beta_{j+1} = \sqrt{(w, z)}$
9: If $\beta_{j+1} == 0$ restart or exit
10: $w_{j+1} := w/\beta_{j+1}$
11: $z_{j+1} := z/\beta_{j+1}$
12: **end for**

It is easy to show that if we set $v_i = Sw_i$, then the v_i's are orthogonal to each other and that they are identical with the sequence of v_i's that would be obtained from the standard Lanczos algorithm applied to $S^{-1}AS^{-1}$ (or $L^{-1}AL^{-T}$) starting with $v_1 = Sw_1$ (or $v_1 = L^{-T}w_1$). The two algorithms are equivalent and going from one to other requires a simple transformation.

The 3-term recurrence now becomes

$$\beta_{m+1} w_{m+1} = \hat{w}_{m+1} = B^{-1}Aw_m - \alpha_m w_m - \beta_m w_{m-1}, \tag{4.13}$$

and $\beta_{m+1} = (B\hat{w}_{m+1}, \hat{w}_{m+1})^{1/2}$. Note that the algorithm requires that we save the auxiliary sequence $z_j \equiv Bw_j$ in order to avoid additional computations with B to calculate B-inner products.

On the surface the extension seems trivial: we could take a sequence of random vectors $w_1^{(k)}$ and compute an average analogue to (4.10) over these vectors. There is a problem in the selection of the initial vectors. We can reason with respect to the original algorithm applied to $S^{-1}AS^{-1}$. If we take a random vector $v_1^{(k)}$ and run Algorithm 2 with this as a starting vector, we would compute the exact same tridiagonal matrix $T_m^{(k)}$ as if we used Algorithm 3 with $w_1^{(k)} = S^{-1}v_1^{(k)}$. Using the same average (4.10) appears therefore perfectly valid since the corresponding $\theta_i^{(k)}$ and $a_i^{(k)}$ are the same. The catch is in the way we select the initial vectors $w_1^{(k)}$. Indeed, it is not enough to select random vectors $w_1^{(k)}$ with mean zero and variance $1/n$, *it is the associated* $v_1^{(k)}$ *that should have this property.* Selecting $w_1^{(k)}$ to be of mean zero and variance $1/n$, will not work, since the corresponding $v_1^{(k)} \equiv Sw_1^{(k)}$ will have mean zero but not the right variance.

The only modification that is implied by this observation is that we will need to modify the initial step of Algorithm 3 as follows:

1. Choose v_1 with components $\eta_i \in \mathcal{N}(0,1)$ and let $w_1 = S^{-1}v_1$ (or $w_1 = L^{-T}v_1$); $z_1 = Bw_1$. Compute $t = \sqrt{(w_1, z_1)}$ and $z_1 := z_1/t; w := w_1/t$. Set $\beta_1 = 0; z_0 = w_0 = 0$.

5. Numerical Experiments. In this section we illustrate the performance of the KPM and Lanczos methods for computing the DOS for generalized eigenvalue problems. Both algorithms have been implemented in MATLAB and all the experiments were performed on a Macbook Pro with Intel i7 CPU processor and 8 GB memory.

In order to compare with the accuracy of the DOS, the exact eigenvalues of each problem are computed with MATLAB built-in function **eig**. We measure the error of the approximate DOS using the relative L_1 error as proposed in [19]:

$$\text{ERROR} = \frac{\sum_i |\tilde{\phi}_\sigma(t_i) - \phi_\sigma(t_i)|}{\sum_i |\phi_\sigma(t_i)|}, \tag{5.1}$$

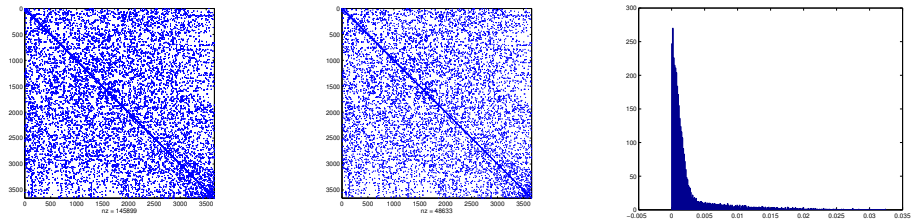
where $\{t_i\}$ are a set of uniformly distributed points and $\phi_\sigma(\cdot)$ and $\tilde{\phi}_\sigma(\cdot)$ are the smoothed (or regularized) DOS with $\delta(t)$ replaced by $\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}}$. A heuristic criterion to select σ as suggested in [20] is to set

$$\sigma = \frac{\lambda_{max} - \lambda_{min}}{60\sqrt{2\log(1.25)}}, \tag{5.2}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of the matrix pencil (A, B).

5.1. An example from the earth's normal mode simulation. The first example is from the study of the earth's normal modes with constant solid materials. The stiffness matrix A and mass matrix B result from the continuous Galerkin finite element method and have size of $n = 3,657$. Details about the physical model and the discretization techniques used can be found in [25, 26].

The numbers of nonzero entries are A and B are $145,899$ and $48,633$, respectively. The eigenvalues of the pencil are ranging from $\lambda_{min} = -2.7395 \times 10^{-13}$ to $\lambda_{max} = 0.0325$. Fig. 5.1 displays the sparsity patterns of A, B as well as the histogram of the eigenvalues of (A, B).



(i) Sparsity of A (ii) Sparsity of B (iii) Histogram of eigenvalues

FIG. 5.1. *For the earth's normal mode matrix pencil, the sparsity pattern of A, B and the histogram of the eigenvalues of (A, B) with 250 bins.*

In Fig. 5.2, we first compare the computed accuracy of the KPM with that of the Lanczos method when the number of random vector n_{nev} was fixed at 50. The Cholesky factorization of B was used for operations involving B. We observe that the Lanczos method outperforms the KPM when m varies from 20 to 60. This is because the eigenvalues of this pencil are clustered near the left endpoint of the spectrum (See Fig. 5.1 (iii)) and the KPM method has a hard time capturing this cluster (See Fig. 5.3).

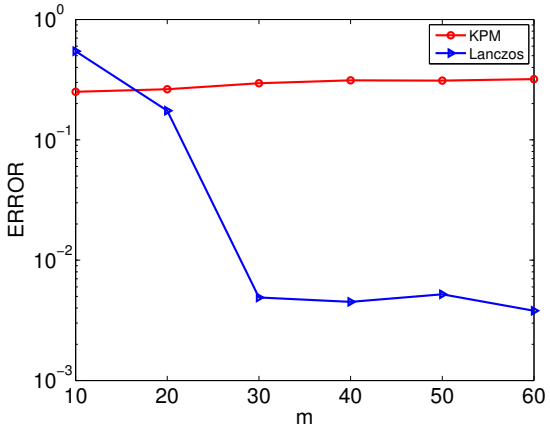


FIG. 5.2. *A comparison of approximation errors of the KPM and Lanczos method applied to the earth's normal mode matrix pencil for different m values. The Cholesky factorization is performed for operations involving B and n_{nev} is fixed at 50.*

Fig. 5.4 shows the error of the Lanczos method with an increasing number of random vectors n_{nev} and fixed $m = 30$. It indicates that a large number of n_{nev} helps reduce the error through the randomization.

Then we consider replacing the Cholesky factorization of B with Chebyshev polynomial approximations $g_{k_1}(B)$ and $q_{k_2}(B)$ as proposed in Section 2.1. One way to determine the degree of g_{k_1} (or g_{k_2}) is to use the theoretical result of Theorem 2.3. However, the theorem has a parameter ρ which is free and the selection of optimal ρ may be harder than the selection of k_i by simpler means. Since $g(t)$, $q(t)$ and their approximations are smooth and a simple heuristic is to select k_i to be the smallest number for which the *computed* $\|(g - g_{k_1})/g\|_\infty$ and $\|(q - q_{k_2})/q\|_\infty$ are small enough. To evaluate the norm we can discretize the interval under consideration very finely

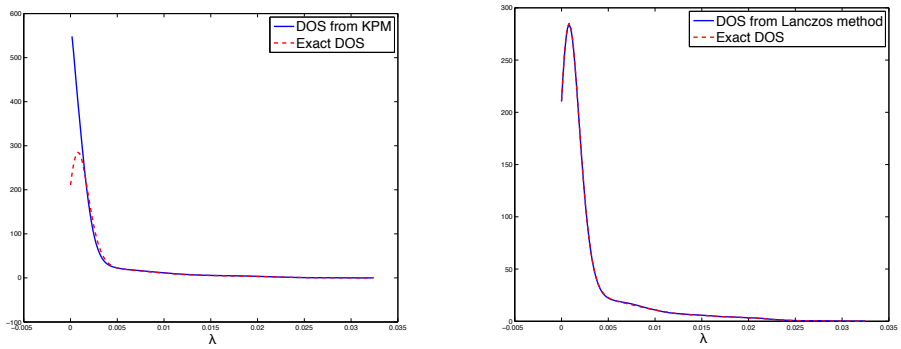


FIG. 5.3. *For the earth's normal mode matrix pencil, the computed DOS by the KPM (left) and the Lanczos method (right) when* $m = 30$ *and* $n_{nev} = 50$*, compared to the exact DOS. Cholesky factorization is performed for operations involving* B*.*

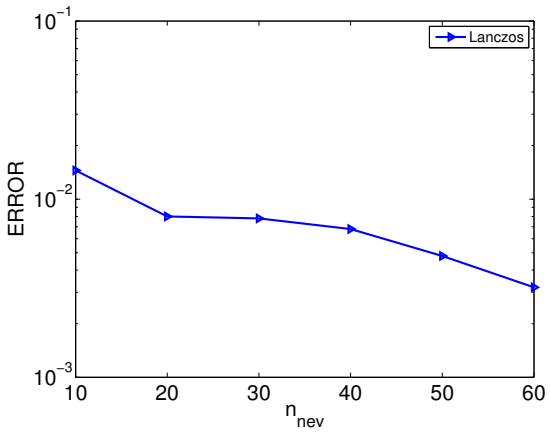


FIG. 5.4. *For the earth's normal mode pencil, the error of the Lanczos method with respect to an increasing number of random vectors* n_{nev} *when* $m = 30$*. Cholesky factorization is performed for operations involving* B*.*

(higher degrees will require more points). This will yield an estimate rather than an exact norm and this is enough for practical purposes.

For the original matrix pencil (A, B), the eigenvalues of B are inside $[3.80e+07, 1.46e+10]$ and $\kappa(B) = 382.91$. In this case, we can estimate the convergence based on $\rho_1 = [\sqrt{\kappa+1} + \sqrt{2}]/[\sqrt{\kappa-1}] = 1.0750$. Since ρ_1 is close to 1, one should expect a slow convergence for $g_{k_1}(B)$ (or $q_{k_2}(B)$) to B^{-1} (or $B^{-1/2}$). In Table 5.1, we report the computed norms $\|(g - g_k)/g\|_\infty$ and $\|(q - q_k)/q\|_\infty$ when k increases from 30 to 60. As we can see, the error associated with g_k is larger than 10^{-3} even when k reaches 60.

We then applied the diagonal scaling technique to the mass matrix B. The eigenvalues of $D^{-1/2}BD^{-1/2}$ are now inside $[0.5479, 2.500]$ and $\kappa(D^{-1/2}BD^{-1/2}) = 4.5629$. In this case, $\rho_1 = 1.9988$ and $g_{k_1}(B)$ and $q_{k_2}(B)$ converge much faster. This is confirmed in Table 5.2 where the error norms are smaller than 6×10^{-6} for both approximations when k reaches 12.

Degree k	$\|(g-g_k)/g\|_\infty$	$\|(q-q_k)/q\|_\infty$
30	8.62×10^{-1}	1.92×10^{-2}
40	3.10×10^{-1}	6.00×10^{-3}
50	1.12×10^{-1}	2.00×10^{-3}
60	4.01×10^{-2}	6.45×10^{-4}

TABLE 5.1

Computed error norms for the order k Chebyshev polynomial approximations to $g=1/\lambda$ and $q=1/\sqrt{\lambda}$ on the interval $[3.8017\times10^7, 1.4557\times10^{10}]$, which contains the spectrum of the original mass matrix B.

Degree k	$\|(g-g_k)/g\|_\infty$	$\|(q-q_k)/q\|_\infty$
6	2.60×10^{-2}	3.73×10^{-4}
8	3.36×10^{-4}	4.32×10^{-5}
10	4.42×10^{-5}	5.13×10^{-6}
12	5.80×10^{-6}	6.19×10^{-7}

TABLE 5.2

Computed error norms for the order k Chebyshev polynomial approximations to $g=1/\lambda$ and $q=1/\sqrt{\lambda}$ on the interval $[0.5479, 2.500]$, which contains the spectrum of the mass matrix B after diagonal scaling.

Fig. 5.5 shows the error of the Lanczos method when the operations $B^{-1}v$ and $B^{-1/2}v$ are approximated by $g_{k_1}(B)v$ and $q_{k_2}(B)v$, respectively. The number of sample vectors n_{nev} was fixed at 50 and the degree m was fixed at 30. The degrees of g_{k_1} and q_{k_2} are determined to be the smallest integers for which the following inequalities hold

$$\|(g-g_{k_1})/g\|_\infty \le \tau, \quad \|(q-q_{k_2})/q\|_\infty \le \tau. \tag{5.3}$$

Although the exact DOS curve is indistinguishable from those obtained from the Lanczos method, the error actually decreases as we reduce the value of τ. The errors are 1.41×10^{-2}, 5.61×10^{-3}, 4.70×10^{-3} and 4.30×10^{-3} when $\tau = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4}, respectively. In the following experiments, we will fix τ at 10^{-3} to select the degree for g_{k_1} and q_{k_2} based on (5.3).

5.2. An example from a Tight-Binding calculation. The second example is from the Density Functional-based Tight Binding (DFTB) calculations (Downloaded from http://faculty.smu.edu/yzhou/data/matrices.htm). The matrices A and B have dimension $n = 17,493$. The matrix A has $3,927,777$ nonzero elements while B has $3,926,405$ nonzero elements. The eigenvalues of the pencil are ranging from $\lambda_{min} = -0.9138$ to $\lambda_{max} = 0.8238$.

Compared with the earth's normal mode matrix pencil, both A, B in this TFDB matrix pair are much denser. Fig. 5.6 displays the sparsity patterns of B and of its Cholesky factor, where nz stands for the number of non-zeros. Even with the help of AMD ordering [1], the number of non-zeros in the Cholesky factor of B still reaches $48,309,857$, which amounts to having 5.5233×10^3 non-zeros per row/column. This will cause two issues. First, a huge amount of memory may be needed to store the factors for a similar problem of larger dimension. Second, applying these factors is also very inefficient. These issues limit the use of Cholesky factorization for realistic large-scale calculations. On the other hand, after diagonal scaling the matrix B has eigenvalues in the remarkably tight interval $[0.5756,\ 1.4432]$, which allows a polynomial of degree

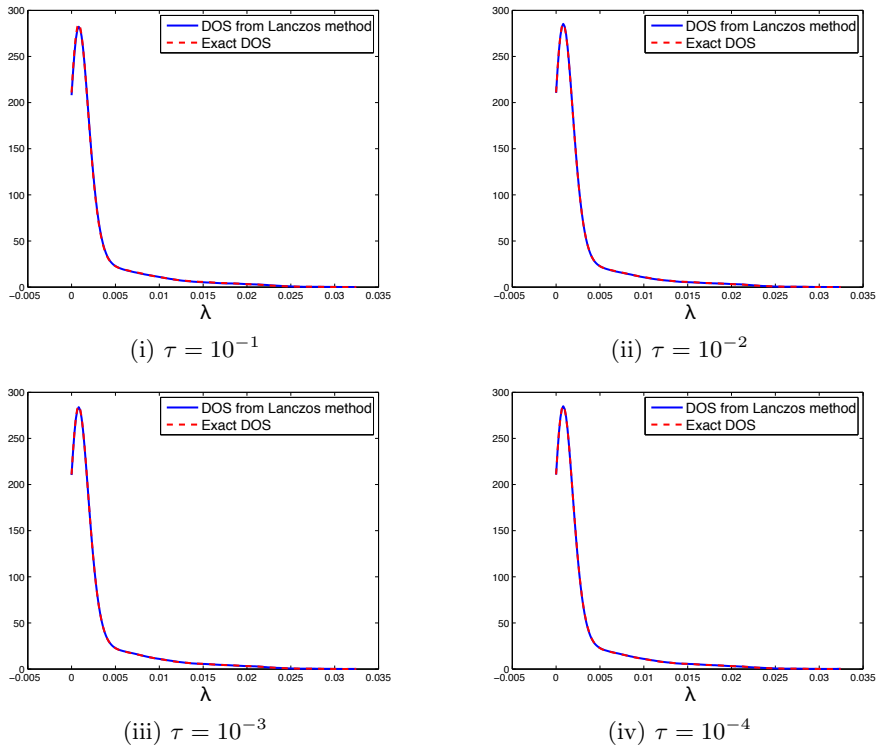


FIG. 5.5. *For the earth's normal mode matrix pencil, the computed DOS by the Lanczos method when* $m = 30$ *and* $n_{nev} = 50$*, compared to the exact DOS. The operations* $B^{-1}v$ *and* $B^{-1/2}v$ *are approximated by* $g_{k_1}(B)v$ *and* $q_{k_2}(B)v$*, respectively. The degrees* k_1 *and* k_2 *are selected to be the smallest integers for which (5.3) hold. The approximation errors are* 1.41×10^{-2}*,* 5.61×10^{-3}*,* 4.70×10^{-3} *and* 4.30×10^{-3} *when* τ *equals* 10^{-1}*,* 10^{-2}*,* 10^{-3} *and* 10^{-4}*, respectively.*

as low as 6 for g_{k_1} and 5 for q_{k_2} when $\tau = 10^{-3}$. Thus, we will only test the KPM and Lanczos method with Chebyshev polynomial approximation techniques for this problem.

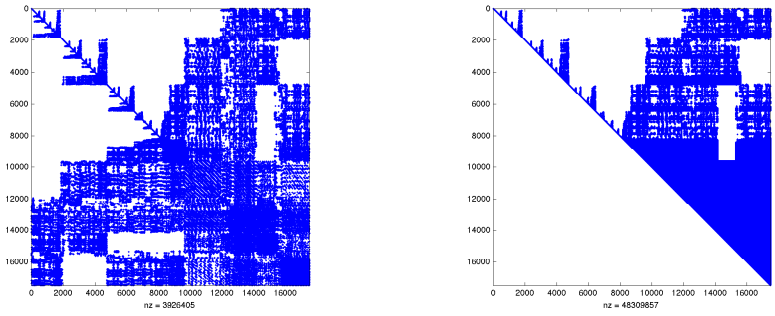


FIG. 5.6. *The sparsity patterns of the matrix* B *(left) and its Cholesky factor (right) for the TFDB matrix pencil. AMD ordering is applied to* B *to reduce the number of non-zeros in its factors.*

In the experiment, we fixed $m = 30$ and $n_{nev} = 50$ in both methods. Fig. 5.7

shows that the quality of the computed DOS by the KPM method is clearly not as good as the one obtained from the Lanczos method. The error for the KPM is 0.2734 while the error for the Lanczos method is only 0.0058. This is because the spectrum of (A, B) has four heavy clusters, which causes difficulties for polynomial-based methods to capture the corresponding peaks on the DOS curve.

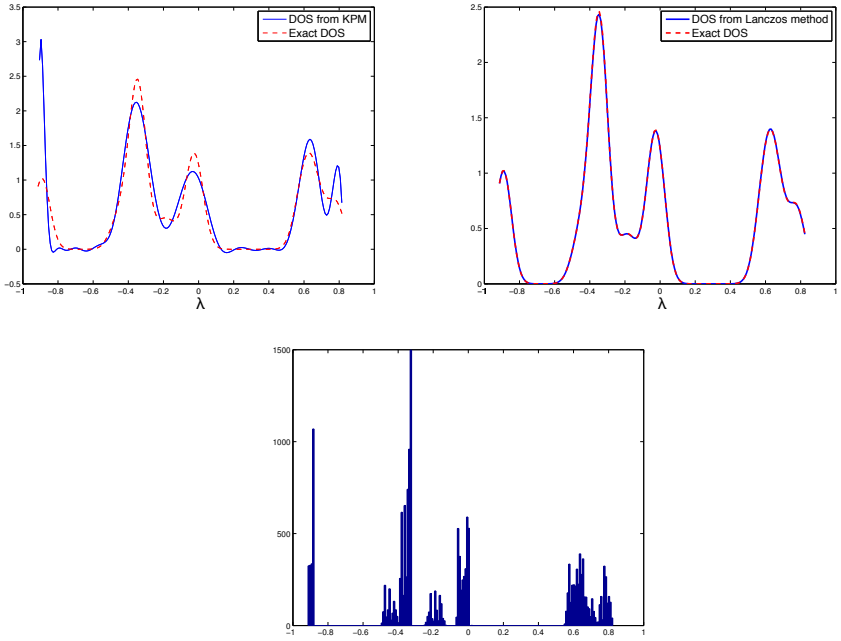


FIG. 5.7. *For the TFDB matrix pencil, the computed DOS by the KPM (upper left) and Lanczos method (upper right) when* $m = 30$ *and* $n_{nev} = 50$*, compared to the exact DOS and the histogram of the eigenvalues with* 200 *bins (lower middle). Chebyshev polynomial approximations are used for operations involving* B*.*

5.3. Application: Slicing the spectrum. This section discusses the spectrum slicing techniques implemented in the EVSL package [9]. First, the Lanczos method is invoked to get an approximate DOS $\tilde{\phi}$ of the input matrix pencil (A, B):

$$\tilde{\phi}(t) = \frac{1}{s}\sum_{k=1}^{s}\sum_{i=1}^{m} a_i^{(k)} g_\sigma(t - \theta_i^{(k)}) \quad \text{with} \quad g_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}}. \tag{5.4}$$

Suppose the users would like to compute all the eigenvalues located inside a target interval $[a,\ b]$ as well as their associated eigenvectors with n_s slices. The interval $[a,\ b]$ will first be finely discretized with $N + 1$ evenly spaced points $x_0 = a < x_1 < \ldots < x_{N-1} < x_N = b$, followed by the evaluation of $\tilde{\phi}_i := \tilde{\phi}(x_i)$ at each point x_i.

Then a numerical integration scheme is used to approximate the following integral based on the computed $\{\tilde{\phi}_i\}$

$$y_i \approx \int_a^{x_i} \tilde{\phi}(t)dt.$$

Each y_i serves an approximation to the number of eigenvalues falling inside $[a,\ x_i]$. In particular, we know there are roughly y_N eigenvalues inside $[a,\ b]$ and should expect an ideal partitioning yielding y_N/n_s eigenvalues per slice.

The endpoints $\{t_i\}$ are identified as a subset of x_i. Start with $t_0 = x_0$. The next t_{i+1} for $i = 0, \ldots, K-2$ is found by testing a sequence of x_j starting with $t_i = x_k$ until $y_j - y_k$ is approximately equal to y_K/n_s, yielding the point $t_{i+1} = x_j$. In the end, the points $\{t_i\}$ separate $[a,\ b]$ into n_s slices.

We illustrate the efficiency of this slicing mechanism with one example. The test problem is to partition the interval $[0.003,\ 0.01]$ into 5 slices for the earth's normal mode matrix pencil. Based on Fig. 5.3, we know that eigenvalues are distributed unevenly within this interval. Therefore, a naive uniform partitioning in which all sub-intervals have the same width will cause some slices to contain many more eigenvalues than others. We fixed m at 30 and varied the number of sample vectors n_{nev} to estimate the DOS for this matrix pencil. The resulting partitioning results are tabulated in Table 5.3. As we can see, even a small number n_{nev} can still provide a reasonable partitioning for the purpose of balancing the memory usage associated with each slice.

	$n_{nev} = 10$		$n_{nev} = 20$		$n_{nev} = 30$	
i	$[t_i,\ t_{i+1}]$	n_i	$[t_i,\ t_{i+1}]$	n_i	$[t_i,\ t_{i+1}]$	n_i
1	[0.0030, 0.0036]	84	[0.0030, 0.0036]	84	[0.0030, 0.0036]	84
2	[0.0036, 0.0045]	90	[0.0036, 0.0045]	90	[0.0036, 0.0045]	90
3	[0.0045, 0.0059]	105	[0.0045, 0.0059]	105	[0.0045, 0.0060]	113
4	[0.0059, 0.0077]	113	[0.0059, 0.0078]	119	[0.0060, 0.0079]	115
5	[0.0077, 0.0100]	110	[0.0078, 0.0100]	104	[0.0079, 0.0100]	98

TABLE 5.3

Partitioning $[0.003,\ 0.010]$ *into* 5 *slices* $[t_i,\ t_{i+1}]$ *for the earth's normal mode matrix pencil. The computational times for the Lanczos method are* $0.53s$, $0.96s$ *and* $1.58s$ *as the number of sample vectors* n_{nev} *increases from* 10 *to* 30. n_i *is the exact number of eigenvalues located inside the* i*th partitioned slice* $[t_i,\ t_{i+1}]$.

6. Conclusion. Algorithms that require only matrix-vector multiplications can offer enormous advantages over those that rely on factorizations. This has been observed for polynomial filtering techniques for eigenvalue problems [10, 18], and it has also just been illustrated in this paper which described two methods to estimate spectral densities of matrix pencils. These two methods use Chebyshev polynomial approximation techniques to approximate the operations involving B and so they only operate on (A, B) through matrix-vector multiplications.

The bounds that were established suggest that the Lanczos method may converge twice as fast as the KPM method under some assumptions and it was confirmed experimentally to produce more accurate estimation when the spectrum contains clusters. The proposed methods are being implemented in C in the EVSL package [9] and will be made available in the next release.

This study suggested that it is also possible to compute eigenvalues and vectors of matrix pairs without any factorization. Theorem 2.6 indicates that rough approximations of the eigenpairs can be obtained by using a low-degree polynomial for $B^{-1/2}$. These approximations can be improved in a number of ways, e.g., by a Rayleigh-Ritz, or a subspace iteration-type procedure. We plan on exploring this approach in our future work.

REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff, *Algorithm 837: An approximate minimum degree ordering algorithm*, ACM Trans. Math. Software, 30 (2004), pp. 381–388.

[2] T. Ando, E. Chow, Y. Saad, and J. Skolnick, *Krylov subspace methods for computing hydrodynamic interactions in brownian dynamics simulations*, J. Chem. Phys., 137 (2012), p. 064106.

[3] E. G. Boman, K. Deweese, and J. R. Gilbert, *An empirical comparison of graph laplacian solvers*, 2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX), (2016), pp. 174–188.

[4] D. Cai, E. Chow, Y. Xi, and Y. Saad, *SMASH: Structured matrix approximation by separation and hierarchy.*, Preprint ys-2016-10, Dept. Computer Science and Engineering, University of Minnesota, Minneapolis, MN, (2016).

[5] J. Chen, M. Anitescu, and Y. Saad, *Computing f(a)b via least squares polynomial approximations*, SIAM Journal on Scientific Computing, 33 (2011), pp. 195–222.

[6] K. Dong and D. Bindel, *Modified kernel polynomial method for estimating graph spectra*, in SIAM Network Science 2015 (poster), May 2015.

[7] D. A. Drabold and O. F. Sankey, *Maximum entropy approach for linear scaling in the electronic structure problem*, Phys. Rev. Lett., 70 (1993), pp. 3631–3634.

[8] A. Weisse, G. Wellein, A. Alvermann, and H. Fehske, *The kernel polynomial method*, Rev. Mod. Phys., 78 (2006), pp. 275–306.

[9] *Eigenvalues slicing library.* http://www.cs.umn.edu/∼saad/software/EVSL/.

[10] H. R. Fang and Y. Saad, *A filtered Lanczos procedure for extreme and interior eigenvalue problems*, SIAM J. Scient. Comput., 34 (2012), pp. A2220–A2246.

[11] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in IN NUMERICAL ANALYSIS, 1994, pp. 105–156.

[12] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rule*, Math. Comp., 23 (1969), pp. 221–230.

[13] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[14] N. Higham, *Functions of Matrices*, Society for Industrial and Applied Mathematics, 2008.

[15] M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Stat. Simul. Comput., 18 (1989), pp. 1059–1076.

[16] V. Kalantzis, R. Li, and Y. Saad, *Spectral schur complement techniques for symmetric eigenvalue problems*, Electron. Trans. Numer. Anal., 45 (2016), pp. 305–329.

[17] L. Kamenski, W. Huang, and H. Xu, *Conditioning of finite element equations with arbitrary anisotropic meshes*, Math. Comput., 83 (2014), pp. 2187–2211.

[18] R. Li, Y. Xi, E. Vecharynski, C. Yang, and Y. Saad, *A Thick-Restart Lanczos algorithm with polynomial filtering for Hermitian eigenvalue problems*, SIAM J. Sci. Comput., 38 (2016), pp. A2512–A2534.

[19] L. Lin, *Randomized estimation of spectral densities of large matrices made accurate*, Numer. Math., 136 (2017), pp. 183–213.

[20] L. Lin, Y. Saad, and C. Yang, *Approximating spectral densities of large matrices*, SIAM Review, 58 (2016), pp. 34–65.

[21] Y. Nakatsukasa, *Absolute and relative weyl theorems for generalized eigenvalue problems*, Linear Algebra Appl., 432 (2010), pp. 242 – 248.

[22] G. A. Parker, W. Zhu, Y. Huang, D.K. Hoffman, and D. J. Kouri, *Matrix pseudospectroscopy: iterative calculation of matrix eigenvalues and eigenvectors of large matrices using a polynomial expansion of the Dirac delta function*, Comput. Phys. Commun., 96 (1996), pp. 27–35.

[23] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, SIAM, Philadelpha, 2011.

[24] G. Schofield, J. R. Chelikowsky, and Y. Saad, *A spectrum slicing method for the kohnsham problem*, Comput. Phys. Commun., 183 (2012), pp. 497 – 505.

[25] J. Shi and M. V. de Hoop, *A note on the parallel computation of earth's normal modes via structured factorization*, Preceedings of the Project Review, Geo-Mathematical Imaging Group, (2016), pp. 223–236.

[26] J. Shi, M. V. de Hoop, R. Li, Y. Xi, and Y. Saad., *Fast eigensolver for computing earth's normal modes*, in Proceedings of the Project Review, Geo-Mathematical Imaging Group, vol. 2, 2017, pp. 317–345.

[27] R. N. Silver and H. Röder, *Densities of states of mega-dimensional Hamiltonian matrices*, Int. J. Mod. Phys. C, 5 (1994), pp. 735–753.

[28] ———, *Calculation of densities of states and spectral functions by Chebyshev recursion and*

maximum entropy, Phys. Rev. E, 56 (1997), p. 4822.
[29] R. N. Silver, H. Röder, A. F. Voter, and J. D. Kress, *Kernel polynomial approximations for densities of states and spectral functions*, J. Comput. Phys., 124 (1996), pp. 115–130.
[30] J. M. Tang and Y. Saad, *A probing method for computing the diagonal of a matrix inverse*, Numer. Lin. Alg. Appl., 19 (2012), pp. 485–501.
[31] L. N. Trefethen, *Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2012.
[32] S. Ubaru and Y. Saad, *Fast methods for estimating the numerical rank of large matrices*, in Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, pp. 468–477.
[33] S. Ubaru, A.-K. Seghouane, and Y. Saad, *Improving the incoherence of a learned dictionary via rank shrinkage*, Neural Computation, (2017), pp. 263–285.
[34] L.-W. Wang, *Calculating the density of states and optical-absorption spectra of large quantum systems by the plane-wave moments method*, Phys. Rev. B, 49 (1994), p. 10154.
[35] A. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.
[36] A. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix.*, Electron. Trans. Numer. Anal., 34 (2008), pp. 125–135.
[37] Y. Xi and Y. Saad, *Computing partial spectra with least-squares rational filters*, SIAM J. Sci. Comput., 38 (2016), pp. A3020–A3045.
[38] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration*, Phy. Rev. E, 74 (2006), p. 066704.