Efficient estimation of eigenvalue counts in an interval

Edoardo Di Napoli * Eric Polizzi † Yousef Saad ‡

August 20, 2013

Abstract

Estimating the number of eigenvalues located in a given interval of a large sparse Hermitian matrix is an important problem in certain applications and it is a prerequisite of eigensolvers based on a divide-and-conquer paradigm. Often an exact count is not necessary and methods based on stochastic estimates can be utilized to yield rough approximations. This paper examines a number of techniques tailored to this specific task. It reviews standard approaches and explores new ones based on polynomial and rational approximation filtering combined with stochastic procedure.

1 Introduction

Recent efforts to develop alternative eigensolvers [21, 15, 20] for large scale scientific applications rely on "splitting" the spectrum of an eigenproblem in intervals and extracting eigenpairs from each one independently. In order to be efficient, this strategy requires an approximate knowledge of the number of eigenvalues included in each of these intervals. The goal of this paper is to explore inexpensive algorithms for determining the number of eigenvalues of a Hermitian matrix that are located in a given interval.

The standard way of computing the number of eigenvalues located inside an interval $[a, b]$ of a Hermitian matrix A is to resort to the Sylvester law of inertia [7]. Note that for the sake of simplicity we limit our discussion to the standard eigenvalue problem in this introduction, although the paper deals with both standard and generalized problems. If A is nonsingular, it admits the decomposition $A = LDL^T$, where L is unit lower triangular, and D is diagonal. The Sylvester inertia theorem then states that the inertias of A and D are the same. This

*Jülich Supercomputing Centre, Forschungszentrum Jülich, D-52425 Jülich. e.di.napoli@fz-juelich.de. Work of this author was supported by VolkswagenStiftung, the Excellence Initiative of the German federal and state governments and the Jülich Aachen Research Alliance – High-Performance Computing.

†Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. polizzi@ecs.umass.edu. Work of this author was supported by the National Science Foundation under Grant #ECCS-0846457.

‡Computer Science & Engineering, University of Minnesota, Twin Cities. saad@cs.umn.edu. Work of this author was supported 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

means that the number of eigenvalues of A that are positive is the same as the number of positive entries in the diagonal of D (Sturm count). Thus, the LDL^T factorizations for the shifted matrices $A - aI$ and $A - bI$ (assuming that these exist) yield respectively the number of eigenvalues larger than a and b. The difference between these two numbers gives the eigenvalue count $\mu_{[a,b]}$ in $[a,\ b]$. While this method yields an exact count, it requires two complete LDL^T factorizations and this can be quite expensive for realistic eigenproblems.

This paper introduces two alternative methods which provide only an estimate for $\mu_{[a,b]}$ but which are relatively inexpensive. Both methods work by estimating the trace of the spectral projector P associated with the eigenvalues inside the interval $[a,\ b]$. This spectral projector is expanded in two different ways and its trace is computed by resorting to stochastic trace estimators, see, e.g.,[8, 23]. The first method utilizes filtering techniques based on Chebyshev polynomials. The resulting projector is expanded as a polynomial function of A. In the second method the projector is constructed by integrating the resolvent of the eigenproblem along a contour in the complex plane enclosing the interval $[a,\ b]$. In this case the projector is approximated by a rational function of A. In either case the eigenvalue count is the trace of P which is estimated via a conventional stochastic technique.

For each of the methods above we present various implementations depending on the nature of the eigenproblem (generalized vs standard), and cost considerations. Thus, in the polynomial expansion case, we propose a barrier-type filter when dealing with a standard eigenproblem, and two high/low pass filters in the case of generalized eigenproblems. In the rational expansion case we have the choice of using an LU factorization or a Krylov subspace method to solve linear systems. The optimal implementation of each method used for the eigenvalue count depends on the situation at hand and involves compromises between cost and accuracy. While it is not the aim of this paper to explore detailed analysis of these techniques, we will discuss various possibilities and provide illustrative examples.

The polynomial and rational expansion methods are motivated by two distinct approaches recently suggested in the context of electronic structure calculations: i) spectrum slicing and ii) Cauchy integral eigen-projection. In the spectrum slicing techniques [21] the eigenpairs are computed by dividing the spectrum in many small subintervals, called ‘slices’ or ‘windows’. For each window a barrier function is approximated by Chebyshev-Jackson polynomials in order to select only the portion of the spectrum in the slice. In this method, it is important to determine an approximate count of the eigenvalue in each sub-interval in order to balance the calculations in a parallel implementation.

The second set of methods is based on eigen-projectors expressed in the form of Cauchy integrals [15, 20]. They essentially compute an orthonormal basis of the invariant eigenspace $\mathcal{V}$ associated with the eigenvalues located in the interval. For these methods to work efficiently one must have a good idea of the dimension of the subspace. This dimension must not be smaller than that of $\mathcal{V}$ if we are to account for all the eigenvalues inside the interval $[a,\ b]$ and, for reasons related to computational costs, it should also not be to large. Thus, we need an estimate that is just slightly larger than $\mu_{[a,b]}$.

The paper is organized as follows. Section 2 introduces the eigenvalue count problem and gives an overview of traditional approaches for solving it. Section 3 discusses methods based on polynomial expansions and Section 4 is devoted to methods based on rational function expansions. Section 5 presents additional issues and provides a series of numerical tests to illustrate the behavior of the method. Finally, Section 6 offers some concluding remarks.

2 Eigenvalue counts

Let $\lambda_j, j = 1, \cdots, n$ be the eigenvalues, assumed here to be labeled by increasing value, and $u_1, u_2, \cdots, u_n$ the associated orthonormal eigenvectors of a Hermitnan matrix A (the generalized problem will be discussed later). The problem we address is to count the number of eigenvalues λ_i in the interval $[a,\ b]$ where we can assume that $\lambda_1 \leq a \leq b \leq \lambda_n$. As was described in the introduction, the standard way of obtaining this count is to resort to the Sylvester inertia theorem which will require two LDL^T factorizations. Let us recall that the inertia of a matrix A is the triplet $n_-(A), n_0(A), n_+(A)$ of its negative, zero, and positive eigenvalues respectively. The number of eigenvalues belonging to the interval $[a,\ b]$ can therefore be computed in one of several ways, for example:

$$\mu_{[a,b]} = n - n_+(A - bI) - n_-(A - aI).$$

Since exact factorizations can be computationally expensive, choosing the correct implementation of LDL^T factorization is crucial. For example in the case of sparse matrices one would like to limit fill-ins in the L factors. In this case the inertias of the matrices $M_a = A - bI$ and $M_b = A - aI$ can be evaluated by leveraging sparse LDL^T factorizations with symmetric pivoting (using 1x1 or 2x2 pivots). Such a sparse factorization computes $PM_iP^T = LDL^T$, where P is a permutation employed to reduce fill-in, and D is a block-diagonal matrix where the blocks are 1x1 or 2x2. Inertias of such block-diagonal matrices are trivial to compute.

For example, in modeling Hamiltonians of 2-dimensional physical systems using finite differences, sparse factorizations can be quite effective and an eigenvalue count based on the inertias may be the method of choice. However, this approach becomes expensive in realistic cases where the matrix arises from simulations of 3-D phenomena. As is well-known [6, 2], in the 3-D case the factorization generates too much fill-in and it is costly both in terms of storage and arithmetic. For dense eigenproblems the number of floating point operations per factorization is of order $O(n^3)$ and this becomes prohibitive for large matrices. Hence, counting eigenvalue based on the inertia theorem is a viable method only when dealing with fairly small dense matrices or large sparse matrices generated by 2-D models.

The problem of eigenvalue counts is also closely related to that of computing "Density of States" (DoS) a term used by physicists for the 'spectral density' or the probability of finding an eigenvalue at a given point in the real line. The spectral density is a distribution, which can be approximated by a smooth function. The eigenvalue count $\mu_{[a,b]}$ is the integral of this distribution over $[a,\ b]$. This viewpoint can help derive methods from techniques for computing density of states. In a recent article [12], a few techniques were developed for approximating spectral densities. Some of these techniques bear some similarity with the ones described here. For example, one can view the polynomial-based techniques presented in this paper as an adaptation of the Kernel Polynomial Method for computing the DoS to the problem of estimating eigenvalue counts.

This paper explores two alternative approaches that compute an estimate of the eigenvalue count in the interval $[a\ b]$ by seeking an approximation to the trace of the eigenprojector:

$$P = \sum_{\lambda_i \ \in \ [a \ \ b]} u_i u_i^T. \tag{1}$$

The eigenvalues of a projector are either zero or one and so the trace of P is equal to the number of terms in the sum (1), i.e., to the number of eigenvalues in $[a,\ b]$. Therefore, we can calculate the number of eigenvalues $\mu_{[a,b]}$ located in the interval $[a,\ b]$ by evaluating the trace of the related projector (1):

$$\mu_{[a,b]} = \mathrm{Trace}(P)\ .$$

If P were available explicitly, we would be able to compute its trace directly and obtain $\mu_{[a,b]}$ exactly. The projector P is typically not available in practice but it is possible to inexpensively approximate it in the form of either a polynomial or a rational function of A. For this, we can interpret P as a step function of A, namely:

$$P = h(A) \quad \text{where} \quad h(t) = \begin{cases} 1 & \text{if } \ t\ \in\ [a\ b] \\ 0 & \text{otherwise} \end{cases} \ . \tag{2}$$

The idea is now to expand $h(t)$ either in a sum $\psi(t)$ of Chebyshev polynomials or to approximate it with a rational function $\chi(t)$, leading to two different approximations of the projector P, namely $P \approx \psi(A)$ or $P \approx \chi(A)$. In this form, it becomes possible to estimate the trace of P by a so-called stochastic estimator developed in the literature [8, 23].

Hutchinson's unbiased estimator uses only matrix-vector products to approximate the trace of a generic matrix A. His idea was based on simulations of the discrete variable which assumes the values -1 and 1 with equal probability 1/2. Thus, to estimate the trace $\mathtt{tr(A)}$ one generates random vectors $v_k, k = 1, .., n_v$ with equally probable entries ± 1 and then computes the average over the sample of $v_k^\top A v_k$

$$\mathtt{tr}(A) \approx \frac{n}{n_v} \sum_{k=1}^{n_v} v_k^\top A v_k. \tag{3}$$

In practice there is no need to take vectors with entries equal to ± 1. Any sequence of random vectors v_k of unit 2-norm will do as long as the mean of their entries is zero [1][1]. Substituting A with either $\psi(A)$ or $\chi(A)$ in (3), we can estimate the trace of P as:

$$\mu_{[a,b]} \approx \begin{cases} \dfrac{n}{n_v} \displaystyle\sum_{k=1}^{n_v} v_k^\top \psi(A) v_k & \text{Polynomial expansion filtering} \\ \dfrac{n}{n_v} \displaystyle\sum_{k=1}^{n_v} v_k^\top \chi(A) v_k & \text{Rational expansion filtering.} \end{cases} \tag{4}$$

The polynomial expansion approach does not require any factorization of A, and this is a big advantage when A is large. Formally, the rational expansion approach would require a few such factorizations, one for each pole z_i of the rational function. However, an exact factorization is no longer needed since we only need to solve linear systems with matrices of the form $A - z_i I$ and preconditioned iterative methods can be invoked for this purpose.

[1] This form was used by physicists to compute the density of states [22, 24, 17, 3, 14].

3 Polynomial expansion filtering

In the polynomial filtering approach, the step function $h(t)$ in (2) is expanded into a degree p Chebyshev polynomial series:

$$h(t) \approx \psi_p(t) = \sum_{j=0}^{p} \gamma_j T_j(t). \tag{5}$$

Here T_j are the j-degree Chebyshev polynomials of the first kind, and the coefficients γ_j are the expansion coefficients of the step function h which are known to be

$$\gamma_j = \begin{cases} \frac{1}{\pi}\left(\arccos(a) - \arccos(b)\right) & : \quad j = 0, \\ \frac{2}{\pi}\left(\frac{\sin(j\arccos(a)) - \sin(j\arccos(b))}{j}\right) & : \quad j > 0. \end{cases}$$

As a result we obtain an expansion of P into matrices $T_j(A)$

$$P \approx \psi_p(A) = \sum_{j=0}^{p} \gamma_j T_j(A). \tag{6}$$

The derivation above is based on the standard assumption that all the eigenvalues of A lie in the interval $[-1,\ 1]$ but it can be trivially extended to a generic spectrum with a simple linear transformation mapping $[\lambda_1,\ \lambda_n]$ into $[-1,\ 1]$. This linear transformation is:

$$l(t) = \frac{t - (\lambda_n + \lambda_1)/2}{(\lambda_n - \lambda_1)/2}$$

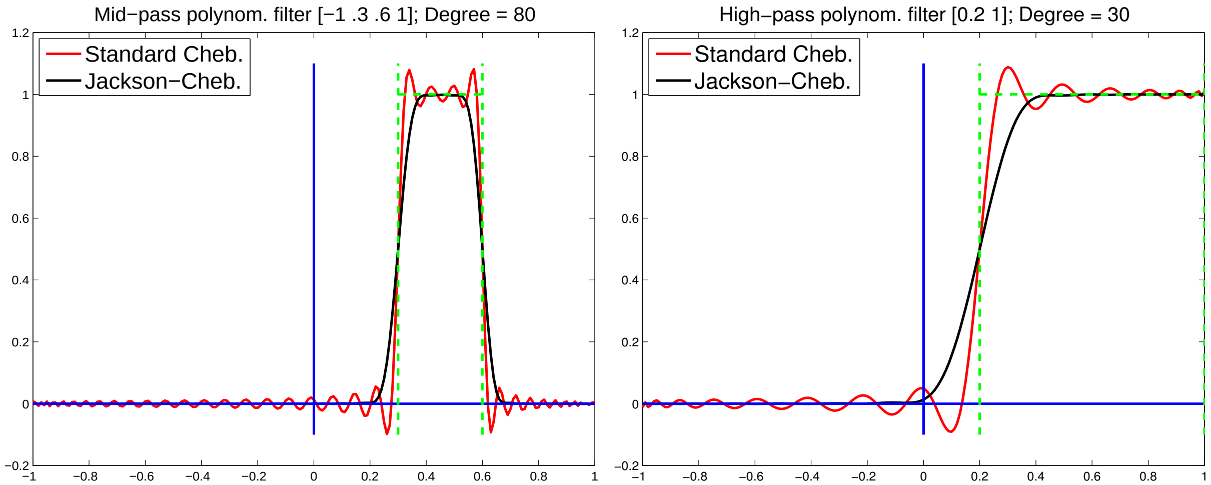
and it requires estimates of the largest and smallest eigenvalues λ_n, λ_1. For the scheme to be stable, it is necessary that the estimate for λ_n be larger than λ_n and the estimate for λ_1 be smaller than λ_1.

Two examples of a Chebyshev expansion of h are shown (in red) in Figure 1. As can be observed from the plots (red curves), the expansion of $h(t)$ has bad oscillations near the boundaries. These are known as *Gibbs oscillations.* To alleviate this behavior it is customary to add damping multipliers – Jackson coefficients – so that (6) is actually replaced by

$$P \approx \psi_p(A) = \sum_{j=0}^{p} g_j^p \gamma_j T_j(A). \tag{7}$$

Notice that the matrix polynomial for the standard Chebyshev approach has the same expression as above with the Jackson coefficients g_j^p all set to one, so we will use the same symbol to denote both expansions. The Jackson coefficients in their original form can be shown to be given by the formula,

$$g_j^p = \frac{\left(1 - \frac{j}{p+2}\right)\sin(\alpha_p)\cos(j\alpha_p) \ + \frac{1}{p+2}\cos(\alpha_p)\sin(j\alpha_p)}{\sin(\alpha_p)} \quad \text{where} \quad \alpha_p = \frac{\pi}{p+2},$$



(a) Mid-pass filter of degree 80.

(b) High-pass filter of degree 30.

Figure 1: Two examples of polynomial filters.

which was developed in [9].

Substituting the expression for $\psi_p(A)$ directly into the stochastic estimator (4), yields the following estimate

$$\mu_{[a,b]} = \text{Trace}(P) \approx \frac{n}{n_v} \sum_{k=1}^{n_v} \left[\sum_{j=0}^{p} \gamma_j v_k^T T_j(A) v_k \right]. \tag{8}$$

A clear advantage of this approach is that it requires only matrix-vector products. In addition, the vectors $w_j = T_j(A)v$ for a given v can be easily computed using the 3-term recurrence relation of Chebyshev polynomials $T_{j+1}(t) = 2tT_j(t) - T_{j-1}(t)$ which leads to

$$w_{j+1} = 2Aw_j - w_{j-1}.$$

An similar approach which uses a more complicated expansion into orthogonal polynomials, was also advocated in [18].

3.1 Generalized eigenvalue problem

We now consider the generalized eigenvalue problem $Ax = \lambda Bx$ where A and B are symmetric and B is positive definite. In this case the projector P in (1) becomes

$$P = \sum_{\lambda_i \;\in\; [a \;\; b]} u_i u_i^T B, \tag{9}$$

and the eigenvalue count is again equal to its trace[2]. However, there are now two matrices involved and this projector does not admit an expression similar to that in (2) for the standard case. A common remedy to this issue is to compute the Cholesky factorization $B = LL^T$ of B and transform the generalized eigenproblem into a standard one with the

[2] Details on this can be found for example in [10, 4].

matrix $L^{-1}AL^{-T}$. This solution reintroduces the need for a costly factorization which we wanted to avoid in the first place. The following simple theorem yields the basis for an efficient alternative:

Proposition 3.1 *Let B be an SPD matrix and $B = LL^T$ its Cholesky factorization. Then the inertias of $A - \sigma B$ and $L^{-1}AL^{-T} - \sigma I$ are identical.*

Proof. It is well-known that the inertias of a matrix C and XCX^T are the same for any nonsingular matrix X, see, e.g., [7]. The proposition follows by applying this result with $C = A - \sigma B$ and $X = L^{-1}$. □

A consequence of the above statement is that we can estimate the number of eigenvalues of the pair (A, B) located in a given interval *without resorting to any factorization.* In essence the idea is to convert the eigenvalue count for the pair (A, B) into two eigenvalue counts for two standard eigenvalue problems. Specifically, we have $\mu_{[a\ b]} = \mu_a - \mu_b$ where μ_a is the number of eigenvalues of $A - aB$ that are positive and μ_b is the number of eigenvalues of $A - bB$ that are positive.

Thus, this approach requires that we expand in the Chebyshev basis the high-pass filters

$$f_\sigma(t) = \begin{cases} 1 & \text{if } t \geq \sigma \\ 0 & \text{otherwise} \end{cases} \qquad f_\sigma(t) \approx \sum_{j=0}^{p} \eta_j^\sigma T_j(t),$$

for $\sigma = a$ and $\sigma = b$ (see Fig. 1b for an example). From these expansions, we would get estimates for the desired counts μ_σ for $\sigma = a, b$ as

$$\mu_\sigma \approx \sum_{j=0}^{p} \eta_j^\sigma \text{Trace}[T_j(A - \sigma B)].$$

The eigenvalue count in the interval $[a,\ b]$ will then be

$$\mu_{[a,\ b]} = \mu_a - \mu_b \approx \frac{n}{n_v} \sum_{k=1}^{n_v} \left[\sum_{j=0}^{p} \eta_j^a v_k^T T_j(A - aB) v_k - \sum_{j=0}^{p} \eta_j^b v_k^T T_j(A - bB) v_k \right].$$

As will be noted in the section devoted to the numerical experiments, for truly generalized problems ($B \neq I$), the spectrum distribution of the matrices $A - \sigma B$ for $\sigma = a, b$, may lead to difficulties, requiring a very large degree polynomial in some cases. It is also possible to count eigenvalues to the left of a and b by using a low-pass filter. Notice that high-pass/low-pass filters require usually a lower degree than mid-pass ('barrier') filters, so this can also be used for the standard eigenvalue problem not just the generalized problem. As for the standard eigenvalue case, costly factorizations are avoided at the expense of using two filters with standard matrices.

4 Rational expansion filtering

A natural extension to the idea of polynomial filtering is to expand P as a rational function. One of several ways of achieving this is via the Cauchy integral definition of a projector:

$$P = -\frac{1}{2i\pi} \int_\Gamma R(z) dz, \tag{10}$$

where $R(z) = (A - zI)^{-1}$ is the resolvent of A, $z \in \mathbb{C}$, and Γ is some smooth curve in the complex plane containing the desired part of the spectrum, see, e.g., [19]. Typically Γ is taken to be a circle whose diameter is the line segment $[a,\ b]$. The above integral is then approximated by resorting to numerical integration methods, leading to

$$P \approx \chi_{n_c}(A) = \sum_{j=1}^{n_c} \gamma_j (A - z_j I)^{-1},$$

where the z_j's are integration nodes and the γ_js are quadrature weights. It is now possible to use the trace estimator by sampling with a set of random vectors. As shown in (4), the trace of P will be approximated by the average of (Pv, v) over many sample vectors v, multiplied by n

$$\mu_{[a,b]} = \text{Trace}(P) \approx \frac{n}{n_v} \sum_{k=1}^{n_v} \left[\sum_{j=1}^{n_c} \gamma_j v_k^T (A - z_j I)^{-1} v_k \right]. \tag{11}$$

4.1 Case of generalized eigenvalue problems

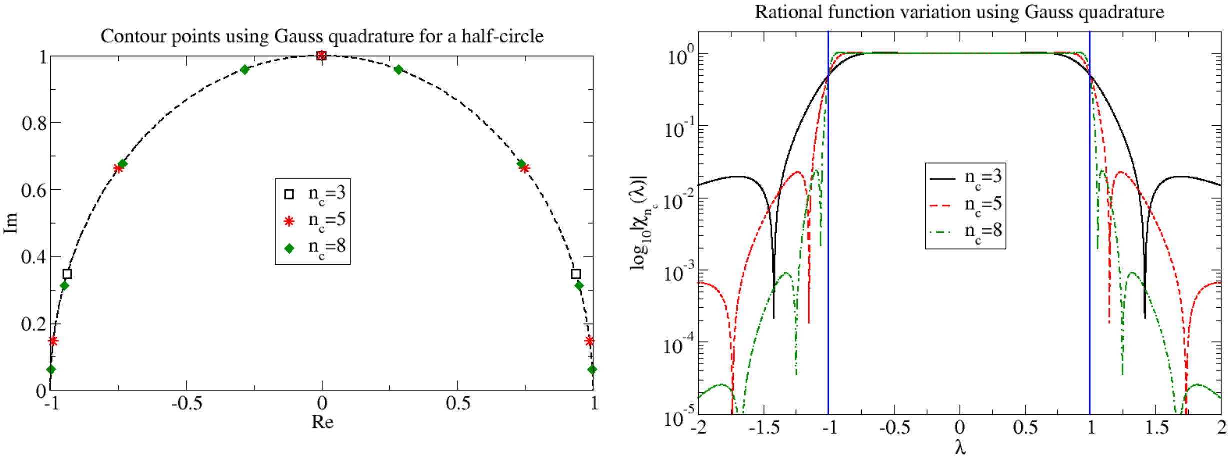
The above formula can be easily adapted to the case of a generalized eigenvalue problem $Ax = \lambda Bx$ where it is assumed here that B is a positive definite matrix. In this case the desired projector is still given by Equation (10), but now the resolvent becomes

$$R(z) = (A - zB)^{-1} B$$

(see [10] for a simple derivation). This means that the only change from the standard case is that the resolvent $(A - z_j I)^{-1}$ in (11) must be replaced by $(A - z_j B)^{-1} B$. In either case the stochastic estimation requires solving linear systems with multiple right hand sides for each integration point. In such situations it is customary to factorize the matrix $A - z_j B$ upfront so as to be able to use the factors repeatedly later. In general such factors do not need to be calculated exactly. For example, when employing an iterative procedure, an approximate factorization of the matrix $A - z_j B$ can be used as a preconditioner for the linear solver.

From the computational cost point of view, this approach may appear to be expensive and not competitive with the one based on Sylvester's inertia theorem described in Sec. 2. Indeed, the inertia approach requires only two factorizations whereas we may now need a few such factorizations to get a good approximation to the spectral projector. In reality the method based on Sylvester's inertia must utilize an exact factorization, whereas in the above formula, all that is needed is to solve linear systems $(A - z_j I)x_k = v_k$ for many right-hand sides v_k by any inexpensive procedure, including iterative methods.

The rational approximation option is particularly appealing in the framework of the FEAST eigensolver [15, 5, 16] (or a related approach). It allows to get a rough eigenvalue count when the factorizations of $A - z_i B$ have been already computed in preparation for a subspace iteration-like procedure. To be specific, the positions of the Gauss-quadrature nodes (i.e. shifts z_i) along the half-circle contour used by FEAST for the symmetric problem are represented on the left of Figure 2. The corresponding values for the rational function applied to λ (i.e. $\chi_{n_c}(\lambda)$) is provided on the right on Figure 2. We note the rapid exponential decay of the rational function from $\simeq 1$ in the middle of the interval $[-1, 1]$ to $\simeq 0$ outside, which both explain the remarkable convergence properties of FEAST subspace iteration



(a) Quadrature nodes.

(b) Rational approximation filter.

Figure 2: Position in the complex plane of the Gauss quadrature nodes for integration along the half-circle, and the corresponding values of the rational function $\chi_{n_c}(\lambda)$ using a semi-log plot. The search interval is here set to $[-1, 1]$.

procedure [4], and the expected efficiency of the trace estimator (11). However, we also note that the eigenvalue count may not be very accurate at the boundaries in case one or both of these boundaries is (are) near clusters of the spectrum (see Section 5.5.2 for more details).

4.2 Implementation issues

Next we consider a few implementation issues related to the rational approximation filtering approach. The eigenvalue count estimator is based on the formula

$$\text{Trace}(P) \approx \frac{n}{n_v} \sum_{k=1}^{n_v} \sum_{j=1}^{n_c} \gamma_j v_k^T (A - \sigma_j I)^{-1} v_k. \tag{12}$$

The above formula involves two loops: the k-loop which we will refer to as the 'sample vector loop' and a j loop which we call the 'integration loop' with n_c the number of contour points in the complex plane. As it is written, the above formula suggests that we would run a vector loop, in which we would generate random vectors, then for each vector in turn we would solve j right-hand sides (integration loop).

In case a direct solver is used for the solutions, this is fine provided we store the factorizations for each integration point. An important observation here is that we can also swap the two loops. In effect we exploit the fact that the trace of the sum of operators is the sum of the different traces:

$$\begin{aligned}\text{Trace}(P) &\approx \text{Trace}\sum_{j=1}^{n_c}\gamma_j(A-\sigma_j I)^{-1}\\ &= \sum_{j=1}^{n_c}\gamma_j\text{Trace}(A-\sigma_j I)^{-1}\\ &\approx \frac{n}{n_v}\sum_{j=1}^{n_c}\gamma_j\sum_{k=1}^{n_v}v_k^T(A-\sigma_j I)^{-1}v_k.\end{aligned}$$

This can be very useful in a processing phase: as each of the n_c factorizations is obtained we generate a number of random vectors and estimate the trace of $(A-\sigma_j I)^{-1}$ with them. At the end of all the factorizations, we end up with an eigenvalue count estimate which can be exploited to determine the subspace dimension to use in the subspace iteration procedure.

Iterative solvers offer an appealing alternative to exact factorizations. When using a Krylov subspace method without preconditioning, one can immediately make the well-known observation that the various systems $(A-\sigma_j I)x_j = v$ for different j's and for each random vector v, can all be solved with the same Krylov subspace (e.g. [13]).

Another issue is to determine what accuracy to require from the solver. This issue was examined in [10] in the context of the package FEAST [5]. The residual norm criterion for GMRES will clearly yield a similar size error for the eigenvector as was observed in [10, 4]. The problem for counting eigenvalues is slightly different. A first observation is that we are not interested in eigenvalues nor eigenvectors. It is well-known that the error on eigenvalues is typically of the order of the square root of the residual norm [19]. From our observations, a high accuracy is not needed. However, there is a minimum accuracy required, below which the method will no longer work. It is also important to have consistent error thresholds. For example, just using a fixed number of GMRES steps will usually not work. It is best to use a criterion based on a residual norm reduction, for example by a factor of 10^{-2}.

5 Numerical experiments

This section provides numerical illustrations of a number of features and discusses additional issues of the methods proposed in Sec. 3 and 4.

5.1 Polynomial filtering: Standard and Jackson polynomials

Our first example is with Matrix 'Na5' from PARSEC. This matrix, available from the University of Florida matrix collection [3], is of size $n = 5832$ and has $nnz = 305630$ nonzero entries. We computed eigenvalues at the outset and defined the interval $[a,\ b]$ so that a is in the middle of $\lambda_{100}, \lambda_{101}$, and b is in the middle of $\lambda_{200}, \lambda_{201}$. In this situation the exact eigenvalue count is 100. Using 30 vectors and a polynomial of degree 70 yields the results shown on the left plot of Figure 3 with standard Chebyshev. The right plot of the same figure shows the result obtained with Jackson-Chebyshev having the same degree polynomial: the

[3] http://www.cise.ufl.edu/research/sparse/matrices/

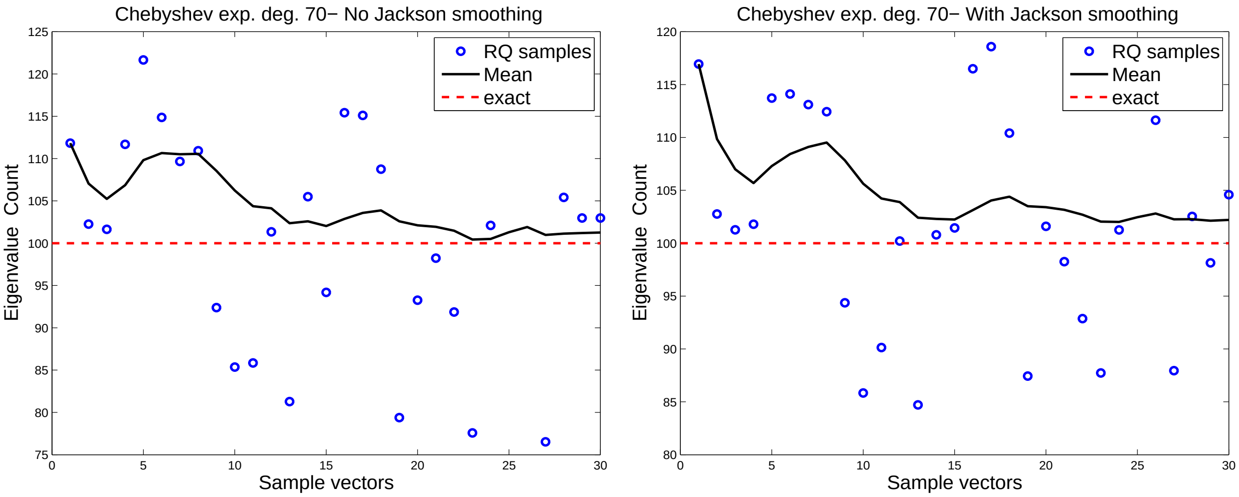


Figure 3: Chebyshev vs Jackson-Chebyshev for counting eigenvalues λ_{101} to λ_{200} for the matrix NA5.

last value of the computed average (eigenvalue count estimate) was 101.25 for Chebyshev and 102.20 for Jackson-Chebyshev. The same sequence of random vectors were used in both Chebyshev and Jackson-Chebyshev.

From this specific example, one may conclude that Jackson tends to often give an overestimate whereas standard Chebyshev often gives an underestimate. In reality the behavior of both methods depends quite crucially from the eigenvalue density distribution in relation to the position of the extrema of the interval $[a\ b]$. In extreme cases their roles are interchanged (see next subsection).

5.2 Estimate bias

In this section we consider only polynomial methods although similar statements can be made for the rational approximation methods. Since $\psi_p(A)$ is only an approximation to the projector (1), its trace will not be equal to the number of eigenvalues inside the interval. In some situations (not involving clustering) a relatively large degree is needed to get a reasonable approximation. An illustration of what can happen when the degree is not large enough is shown in Figure 4. On the left side a low degree polynomial is used. The lower horizontal line is the actual trace of the matrix $\psi_p(A)$. So the trace estimator does work, but it estimates a trace of an inaccurate projector. The higher dashed horizontal line is the actual eigenvalue count. As can be seen, there is a substantial gap between the two. The right side of the figure shows that the gap narrows substantially for a higher degree polynomials. In this case, a degree above 70 is necessary to get an approximation that is close enough, where the lower and upper dashed lines are close. In this regard there could be a big difference between the Jackson-Chebyshev and the standard Chebyshev polynomials as is shown in Table 1. Here Jackson smoothing seems to be very detrimental to the estimation. In other situations, the Jackson polynomial performs better.

We explored a few ways to fix this bias. In particular the simplest correction is to compare

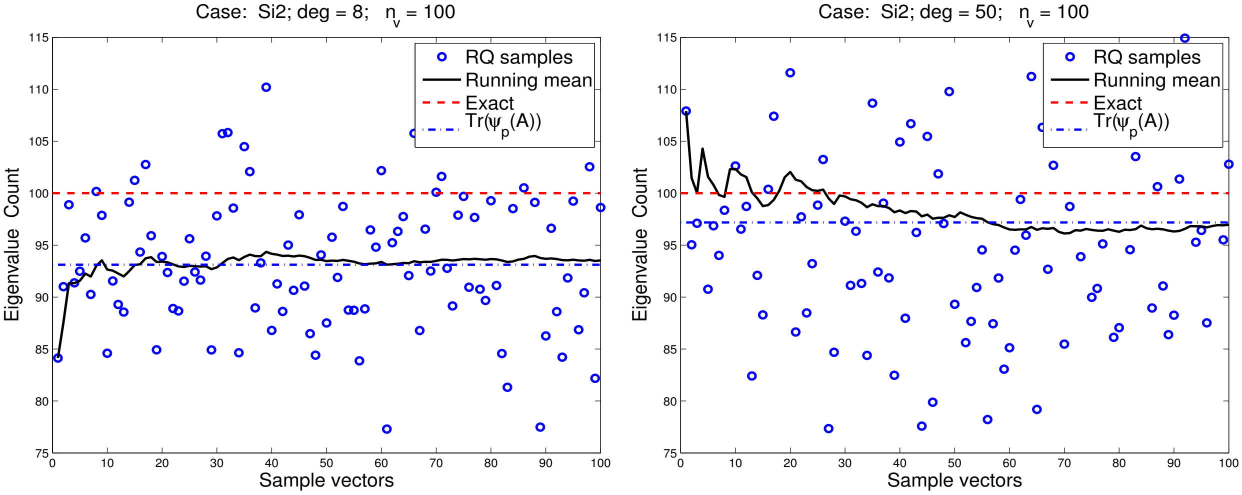


Figure 4: Chebyshev based counting with a degree 8 (left) and 50 (right) for matrix Si2.

p	8	20	30	40	50	70	100	120
tr $[\psi_p^S(A)]$	93.12	97.29	96.98	96.81	97.19	101.58	101.54	100.76
tr $[\psi_p^J(A)]$	74.53	89.59	93.00	94.51	95.29	95.97	96.99	97.74

Table 1: Evolution of the trace of $\psi_p(A)$ for the standard Chebyshev ($\psi_p^S(A)$) and Jackson-Chebyshev ($\psi_p^J(A)$) approaches for the $Si2$ test case. The exact count is 100.

the integrals of $\psi_p(t)$ in $[-1,\ 1]$ with that of the step function on the same interval. The integral of the step function is just $b-a$. The integral of each T_k in the interval $[-1,\ 1]$ is readily computable (it is equal to zero when k is odd and to $-2/(k^2-1)$ when k is even). Then one can obtain the integral of ψ_p in $[a,\ b]$ from which a corrective factor can be obtained. However, experiments with such a correction were mixed. The difficulty inherent to this problem is that the bias will certainly depend on the distribution of eigenvalues.

5.3 Rational approximation filtering: Direct vs iterative solvers

The matrices from PARSEC are relatively small but they simulate three-dimensional phenomena. In addition, the number of nonzeros per row is relatively high. This leads to rather expensive exact factorizations for the larger problems.

In Figure 5, we ran the same Na5 example that was tried before with Chebyshev polynomials, using direct factorizations for each contour point. The final average obtained using n_v=40 sample vectors, is 98.64 for 3 contour points and 100.27 for 5 contour points. Using more contour nodes can indeed systematically improve the result of the count. It is important to note that much faster results can be obtained using Sylvester and the inertia approach since only two direct factorizations (in real arithmetic) are needed.

In order to improve the performance of the rational approximation as well as its applica-

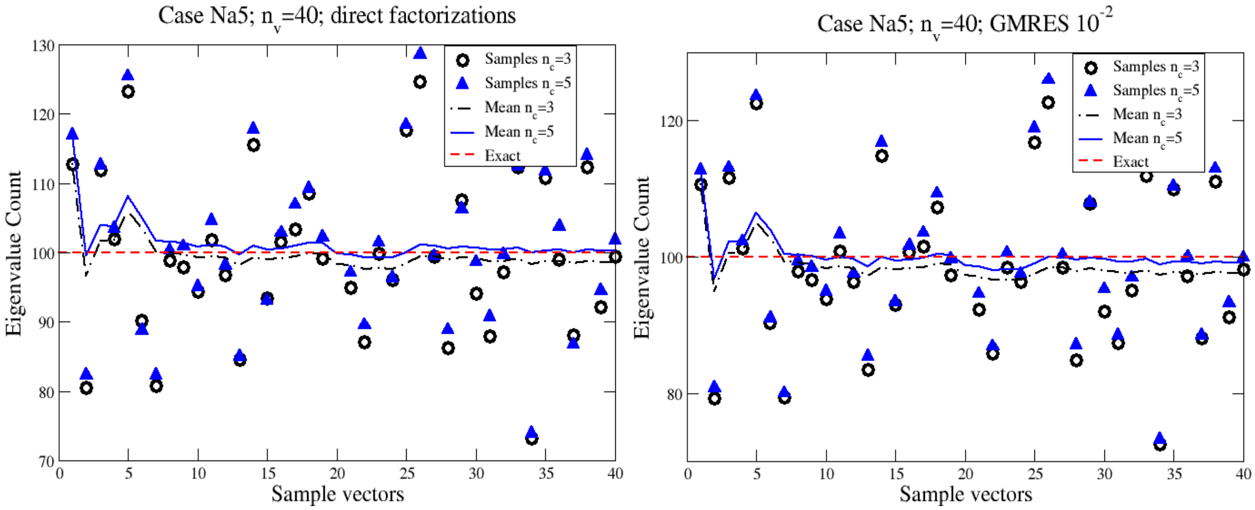


Figure 5: The rational approximation method at work for the Na5 example using a direct solver (left) and an iterative solver (right). The plots compare runs using 3 and 5 Gauss contour points for integration along 40 sample vectors.

bility to large sparse systems, we can avoid the direct factorizations and make use of iterative solvers instead. The right side of Figure 5 presents the Na5 example using GMRES with a modest criteria of convergence of 10^{-2}. We note that the estimates are reasonably good in comparison with the direct cases presented in Figure 5 using only few sample vectors. Using $n_c = 3$ contour points and $n_v = 40$ sample vectors, the last average is 97.61. Two other runs using 10^{-1} and 10^{-3} for the GMRES stopping criteria, lead to a final average of 75.35 and 98.65 respectively. As mentioned above a minimum accuracy for GMRES is necessary. Using $n_c = 3$, Figure 6 provides the absolute errors on the estimates of the eigenvalue counts obtained by comparing direct factorizations and GMRES with 10^{-2} and 10^{-3} convergence criteria. This latter is in extremely good agreement with the direct factorization results, and such agreement is expected to be preserved independently of the number of contour points. In the absence of a preconditioner, we note that the iterative procedure requires only matrix-vector multiplication operations from the matrix A. Therefore, this is a promising approach for handling very large systems. In order to further enhance the performances, one can think of two options to be developed: (i) the generation of a single Krylov subspace common to all contour points; (ii) the use of (cheap) preconditioners.

5.4 Eigenvalue count estimates for the FEAST eigensolver

An estimate of the number of eigenvalues in the search interval $[a,\ b]$ can provide useful information for setting up a lower bound on the subspace size for the FEAST eigensolver. In most cases, a subspace size of $\times 1.5$ to $\times 2$ the number of eigenvalues, is needed by FEAST to provide a good convergence rate using the Gauss-quadrature rule for the contour integration. Within the subspace iteration framework of FEAST [4], it is useful to note that the convergence rate of the eigensolver is directly dependent on the value of the rational function at $a - \alpha$ and $b + \alpha$ where $\alpha > 0$. In Figure 2, we note for the case $n_c = 8$ and the search interval $\lambda \in [-1, 1]$ that the rational function is equal to $\sim 10^{-4}$ for $\lambda = \pm 1.5$. As

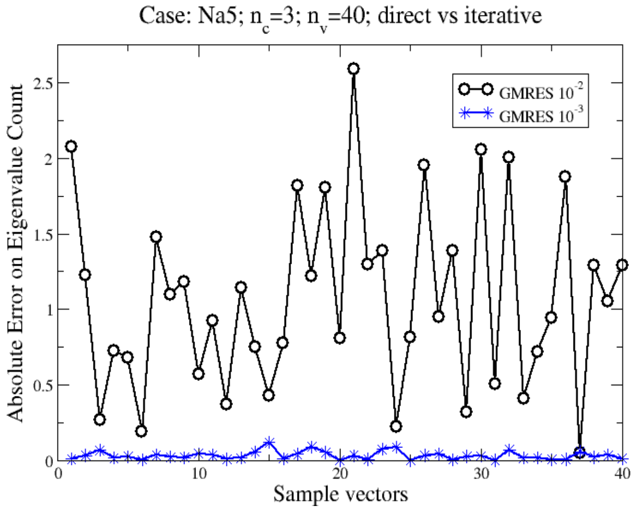


Figure 6: Absolute relative errors on the eigenvalue count for the Na5 example using 3 contour points while comparing direct factorization and GMRES with both 10^{-2} and 10^{-3} convergence criteria.

a result, if the search subspace size is taken large enough to include all eigenvalue counts between $[-1.5, 1.5]$, one can expect the eigenpairs residuals within $[-1,\ 1]$ to converge with the same (linear) rate of 10^4 as for the FEAST subspace iterations (while the eigenvalues should converge with a (linear) rate of $(10^4)^2 = 10^8$). Stated otherwise, if $n_c = 8$ contour points are to be employed by the FEAST eigensolver along with a search interval $[a,\ b]$, a good estimate of the number of eigenvalues within $[a-\alpha,\ b+\alpha]$ with $\alpha = (b-a)/4$, should then guarantee a convergence rate of 10^4 (excluding the event that no eigenvalues can be found close to the edges of the new larger interval). In turn, this eigenvalue count within the larger interval can be estimated using a reduced number of contour points that can take advantage of iterative solvers with modest residuals as presented in the previous section.

	$n_c = 8$			$n_c = 5$		
Iteration	# eigenvalue	residual	conv. rate	# eigenvalue	residual	conv. rate
0	62	1.96×10^{-1}		65	2.06×10^{-1}	
1	59	1.58×10^{-7}		59	1.18×10^{-3}	
2	59	8.78×10^{-11}	1.8×10^{4}	59	3.89×10^{-5}	3.0×10^{1}
3	59	1.46×10^{-14}	6.0×10^{4}	59	6.48×10^{-8}	6.0×10^{2}
4				59	1.30×10^{-11}	5.0×10^{3}
5				59	5.02×10^{-14}	2.6×10^{2}

Table 2: Convergence results obtained using FEAST v2.1 for a new interval $[a,\ b]$ containing 59 eigenvalues and where the subspace size has been estimated at 100 by counting the eigenvalues between $[a-(b-a)/4,\ b+(b-a)/4]$. Convergence rates are provided from the moment the number of found eigenvalues found stabilizes.

In the following, we illustrate how the new appropriate estimate on a larger search interval can be used by FEAST to guarantee a certain degree of convergence rate. In order to make use of the same results obtained previously for Na5, let us assume that the 100 exact eigenvalues and the various estimates provided, were actually representing the number of eigenvalues within an interval $[a-(b-a)/4,\ b+(b-a)/4]$, where $a-(b-a)/4 = 1.2629..$ and $b+(b-a)/4 = 2.00689...$ We propose to define a new search interval of interest $[a,\ b]$ with a, b being the original interval bounds $a = 1.38695..$ and $b = 1.88290..$, as a starting point for the FEAST eigensolver also now using the estimated subspace size of 100. Table 2 represents the convergence results obtained by FEAST v2.1[5] using 5 and 8 contour points. These results show a linear convergence rate for $n_c = 8$ in agreement with the expected 4×10^4 that can be directly obtained from a reading of the data in Figure 2 (at $\lambda = \pm 1.5$). On the other hand, the value of the rational function using $n_c = 5$ indicates an expected convergence rate of 2×10^2 while the reported rates in Table 2 appears to be in agreement or much better after few iterations.

5.5 Eigenvalue distribution and accuracy

The a priori determination of the polynomial degree and the number of nodes for a specific requested accuracy of the eigenvalue estimates is a difficult task. In this section we address this issue in relation with the presence of clusters of eigenvalues in the proximity of the eigenvalue interval $[a,\ b]$.

5.5.1 Polynomial expansion

We saw in Section 5.2 that the accuracy of the count estimate by the polynomial method can be improved by increasing the degree of the polynomial. While this is a reasonable expectation, in general the accuracy of the eigenvalue estimate depends on the eigenvalue distribution of A. In particular if either a or b is close or inside a cluster of eigenvalues, $\mu_{[a,\ b]}$ may overestimate or underestimate the true number of eigenvalues in $[a,\ b]$. Moreover the extent of the error is related to both the size of the cluster and the relative distance between eigenvalues in the cluster $D_\lambda = \frac{\|\lambda_{i+1}-\lambda_i\|}{\|\lambda_i\|}$.

In order to illustrate and quantify this point we ran a series of tests on an eigenproblem containing two artificially engineered clusters of distinct length and density. For each cluster we placed the rightmost ending of the interval $[a,\ b]$ either at the beginning or at the end of the set of values forming the cluster. For each case we computed $\mu_{[a,\ b]}$ using both the simple Chebyshev $\psi_p^S(A)$ and the Jackson-Chebyshev expansion $\psi_p^J(A)$ with two quite different polynomial degrees p. The results are summarized in Table 3 where overestimation and underestimation are indicated by an line over and under the measured number respectively.

Results from the larger and denser cluster of eigenvalues – $D_\lambda = 2 \times 10^{-9}$ – clearly show that when the filtering interval intersects the beginning of a cluster both $\psi_p^S(A)$ and $\psi_p^J(A)$ overestimate the eigenvalue count. In contrast when the interval intersects the end of a cluster both expansions underestimate the eigenvalue count. This result is only mildly dependent on the polynomial degree adopted and reflects the tailing effects of the expansions as shown for example in Fig. 1a. In that figure, corresponding to $[a,\ b] = [0.3,\ 0.6]$, the polynomial expansion either undergoes strong oscillation (Chebyshev) or a damping effect (Jackson-Chebyshev). In either case this tailing effect adds to the estimate when the cluster is adjacent but mostly outside the interval, and it subtracts from the estimate when the cluster is adjacent but mostly inside the interval.

The extent of the error made depends on the length of the cluster and on the relative distance between eigenvalues in the cluster. In fact, for the smaller and sparser cluster with $D_\lambda = 5 \times 10^{-5}$, both estimates have a far smaller error and actually succeed to not overestimate the eigenvalue count for the lower degree. Once more this conclusion is almost independent from the polynomial degree as long as this degree is not too small.

Cluster length	D_λ	$\mathrm{N_v}$	Poly degree	b position w.r.t. cluster	Trace of $\psi_p^S(A)$	Trace of $\psi_p^J(A)$	Exact
13	5×10^{-5}	50	30	Beginning	$\underline{46.48}$	$\underline{45.67}$	48
				End	$\underline{47.35}$	$\underline{46.39}$	55
			100	Beginning	$\overline{52.13}$	$\overline{49.53}$	48
				End	$\underline{53.17}$	$\underline{50.91}$	55
33	2×10^{-9}	50	30	Beginning	$\overline{99.80}$	$\overline{94.85}$	90
				End	$\underline{100.98}$	$\underline{95.54}$	116
			100	Beginning	$\overline{102.54}$	$\overline{99.63}$	90
				End	$\underline{104.09}$	$\underline{100.53}$	116

Table 3: Trace estimates for the polynomial approximation in the presence of a cluster of eigenvalues closer to the rightmost end of the interval $[a,\ b]$. The labels “Beginning” and “End” refer to the case when most of the values of the cluster are respectively outside or inside the interval. Numbers having a line over or under them indicate that the trace either overestimates or underestimates the corresponding exact count.

5.5.2 Rational expansion

Similar to polynomial expansions, the rational expansion of the projector is also sensitive to the spectrum distribution near the extrema of the interval $[a,\ b]$. We repeated a set of tests similar to those described in Section 5.5.1 and collected the results in Table 4.

For the rational expansion $\chi_{n_c}(A)$ the role played by the polynomial degree in estimating the trace of $\psi_p(A)$ is taken on by the number n_c of integration nodes. While the dependence of the eigenvalue count on the polynomial degree was mild, in this case as soon as n_c exceeds 5 the trace of the projector does not undergo substantial changes. This behavior is mainly due to the rapid decay of the error of the Gauss quadrature method. Despite the extremely stable behavior of the quadrature, $\mu_{[a,\ b]}$ still shows the same inaccuracies observed for the polynomial expansion $\psi_p(A)$ when facing clusters of eigenvalues. When most of the values characterizing a cluster are close but outside the interval $[a,\ b]$, the trace of $\chi_{n_c}(A)$ overestimates the number of eigenvalues. In contrast, if such cluster of values is almost entirely within the filtering interval, the eigenvalue count is underestimated.

As for the case of the polynomial expansion, these results could be explained by considering the influence on the tailing effects of the projector as shown in Fig. 2b. Thus, if the cluster is just to the right of b, the trace estimator will evaluate the trace of an operator $\chi_{n_c}(A)$ that includes a large number (cluster) of not so small terms $\chi_{n_c}(\lambda_i)$ added to this trace, resulting in an upward bias, i.e., an overestimate of the trace. In other words, clusters located near the end-points will just exacerbate the bias mentioned in section 5.2.

Cluster length	D_λ	$\mathrm{N_v}$	Integration nodes n_c	b position w.r.t. cluster	Trace of $\chi_{n_c}(A)$	Exact
13	5×10^{-5}	50	5	Beginning	$\overline{50.86}$	48
				End	$\underline{51.19}$	55
			16	Beginning	$\overline{49.94}$	48
				End	$\underline{51.07}$	55
33	2×10^{-9}	50	5	Beginning	$\overline{100.62}$	90
				End	$\underline{98.45}$	116
			16	Beginning	$\overline{101.75}$	90
				End	$\underline{100.57}$	116

Table 4: Trace estimates for the rational approximation in the presence of a cluster of eigenvalues closer to the rightmost end of the interval $[a,\ b]$. The labels "Beginning" and "End" refer to the case when most of the values of the cluster are respectively outside or inside the interval. Numbers having a line over or under them indicate that the trace either overestimates or underestimates the corresponding exact count.

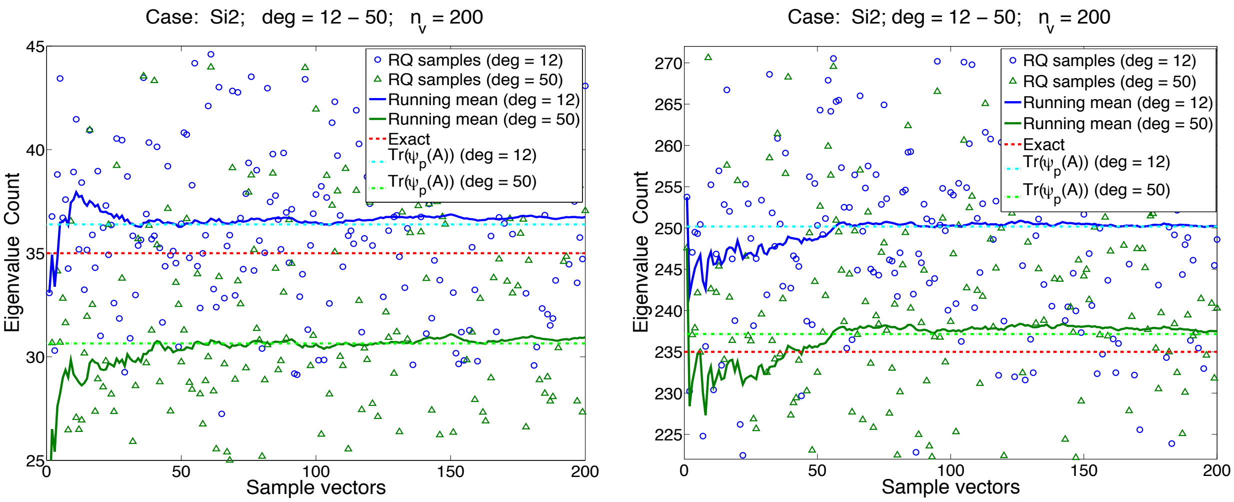
5.6 Number of sample vectors and convergence of the trace

The approach advocated in this paper relies on a relatively inexpensive method to compute the trace of eigen-projectors using a stochastic estimator which uses a certain number of sample vectors. It is interesting to investigate what minimum number of sample vectors is needed for the expansion to converge to a stable estimate. In the following we provide a general set of observations made in our numerical tests.

5.6.1 Polynomial expansion

In the polynomial expansion the degree of the polynomial mainly influences the accuracy with which $\psi_p(t)$ approximates the function $h(t)$. When computing the trace of $h(A)$ the degree of the polynomial exercises this influence by aligning each vector v_k along the direction of the eigenvectors corresponding to the eigenvalues in $[a,\ b]$: the higher the polynomial degree the more accurate such an alignment is. In order to have a significant representation of all eigenvectors directions the total number of vectors needs to be higher of a certain minimum.

In our tests we observed that in most cases, even for intervals containing a small number of eigenvalues, a minimum number of 40 vectors may be needed (see Fig. 7a). When the number of eigenvalues in $[a,\ b]$ is larger it may be unavoidable to increase the number of



(a) Small interval $[a,\ b]$ including λ_5 up to λ_{40}. (b) Large interval $[a,\ b]$ including λ_5 up to λ_{240}.

Figure 7: Comparison of trace estimate for two distinct intervals and two different polynomial degrees for the Si2 case with a total of 200 vectors. In each plot the random generated vectors are the same for both estimates.

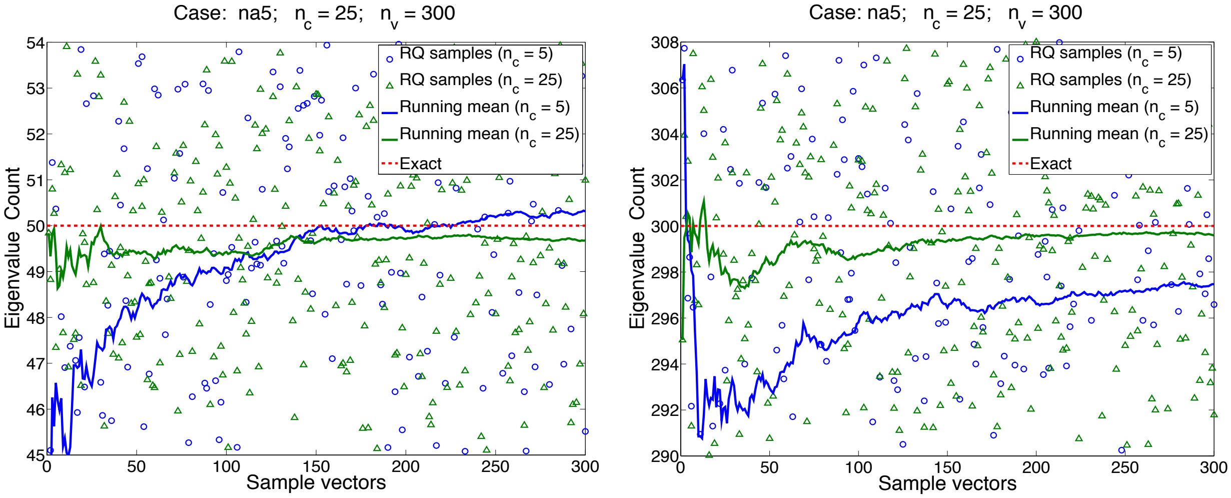
vectors in order to have a reliable estimate of the trace of the projector P. In general, this increase is not linear but somewhat proportional to the square root of the increment. In other words more than 40 vectors are needed but not many more.

This situation is clearly depicted by Fig. 7. In plot 7a the estimate is clearly stabilized around its average value Trace($\psi_p(A)$) for a number of vectors greater than 40. In plot 7b the estimate stabilizes for a number of vectors larger than 60. As is seen in both plots the degree of the polynomial determines the converged value of the estimate (it lowers it quite substantially in this specific case) but does not influence the threshold $s_{\rm poly}(v)$ at which the stochastic estimate starts to converge: $s_{\rm poly}(v)$ depends mainly on the number of eigenvalues in the interval. For the polynomial expansion this dependence seems to be not too strong.

5.6.2 Rational expansion

In the rational expansion case, the accuracy of the trace of P relies on the number of integration nodes n_c. As for the polynomial expansion, the minimum number of vectors $s_{\rm ratio}(v)$ for which the computed trace starts to converge is not that sensitive to the number n_c of poles of the rational function used. Rather, it is affected by the number of eigenvalues in $[a,\ b]$. This is illustrated in Fig. 8 where $s_{\rm ratio}(v) \sim 50$ for an interval containing 50 eigenvalues (see plot 8a) while $s_{\rm ratio}(v) \sim 150$ for an interval with 300 eigenvalues (see plot 8b). In practice when the number of eigenvalues in the interval increases six-fold, $s_{\rm ratio}$ becomes 3 times larger. In other words $s_{\rm ratio}$ seems to depend on the dimension of the range of P.

While this conclusion seems natural due to the nature of the rational expansion, it may introduce a practical difficulty in estimating a priori the minimum number of vector necessary for the estimation of the trace. This difficulty can be overcome if the order of magnitude of the number of eigenvalues in $[a,\ b]$ is known in advance. Unfortunately, increasing the number



(a) Small interval $[a,\ b]$ including λ_{100} up to λ_{150}. (b) Large interval $[a,\ b]$ including λ_{100} up to λ_{400}.

Figure 8: Comparison of trace estimate for two distinct intervals and two different choices of integration nodes for the Na5 case with a total of 300 vectors. In each plot the random generated vectors are the same for both estimates.

of nodes does not soften the dependence of s_{ratio} on $\mu_{[a,\ b]}$ as can be seen by comparing the blue and green lines in both plots of Fig. 8. Seen from this angle, the polynomial expansion approach has an intrinsic advantage over the rational expansion method.

5.7 Tests with generalized problems

We tested the polynomial and rational filtering methods on generalized eigenvalue problem issued from a 2-D FEM simulation matrix and available from the FEAST package [5]. The matrices A, B have size $n = 12,450$. The number of nonzero entries in both A and B is $nnz = 86,808$. The number of eigenvalues inside the desired interval is 100 (i.e. 100 lowest eigenvalues). The left side of Figure 9 shows the result obtained with polynomial filtering when the degree is 100.

We also ran the rational filtering method with 5 integration points, using a direct solver and GMRES with and without preconditioning (denoted by GMRES and P-GMRES respectively). The results shown on the right side of Figure 9 will allow a comparison of the these approaches. To improve clarity the figure shows only the running averages and omits the small circles corresponding to the Rayleigh quotients of each sample. The direct factorization used is the one provided by MATLAB with its default reordering. We also tested preconditioned GMRES with an ILU factorization using a drop tolerance of `droptol=0.01` and a pivoting threshold of 0.05 as defined by the `ilu` function from MATLAB . For all the runs with (P)-GMRES we used a restart dimension of $m = 20$ and limited the number of steps to 200. The iteration is stopped as soon as the residual norm drops by a certain convergence tolerance (`tol`). We used two values for `tol`: 1.e-03 and 1.e-02.

In our numerical experiments the results obtained using GMRES - with and without preconditioning - when using a convergence tolerance of 10^{-3} for the residual, are quasi-identical with the ones provided by the direct approach which are optimal for a fixed n_c.

Accordingly, the results with (P)-GMRES (`tol=1.e-03`) are not shown.

Instead, we show the results obtained using (P)-GMRES with `tol=1.e-02`. The curves shows an underestimation of the eigenvalue counts for both GMRES and P-GMRES. Remarkably, these two curves are consistently below that of the direct (or GMRES(`tol=1.e-03`) by a number which does not deviate too much from the average difference which is ≈ 8.7 for the non-preconditioned case and ≈ 4.36 for the preconditioned case. This indicates the existence of a bias for both GMRES and P-GMRES as well, for the case of insufficient accuracy. This observation is not easy to explain and deserves further investigation. Note that for the purpose of comparisons, the sequence of sample vectors used in all experiments described above (including those with polynomial filtering) are identical.

Though the rational approximation approach delivers more accurate results it is more costly in terms of memory. One must take into account the number of nonzero elements for all the LU factorizations generated for the 5 integration points. This number for this case is $nnz_{LUtotal} = 22,239,530$, when $n_c = 5$, an enormous amount relative to the original number of nonzero entries which is under $100,000$. It is reduced to $nnz_{LUtotal} = 13,343,718$, for $n_c = 3$. In contrast, when the ILU factorization is used for $n_c = 5$ and under the conditions of the above experiment the total number of nonzero elements used for all 5 ILU factorizations drops to $nnz_{LUtotal} = 799,143$, or about 28 times smaller than the number of nonzero entries required by the corresponding direct solver.

This is a 2D problem. For a 3D problem with which we have made a few tests a direct approach becomes infeasible on a standard workstation. It then becomes essential to use a preconditioned iterative solver with a well-tailored incomplete factorization preconditioner.

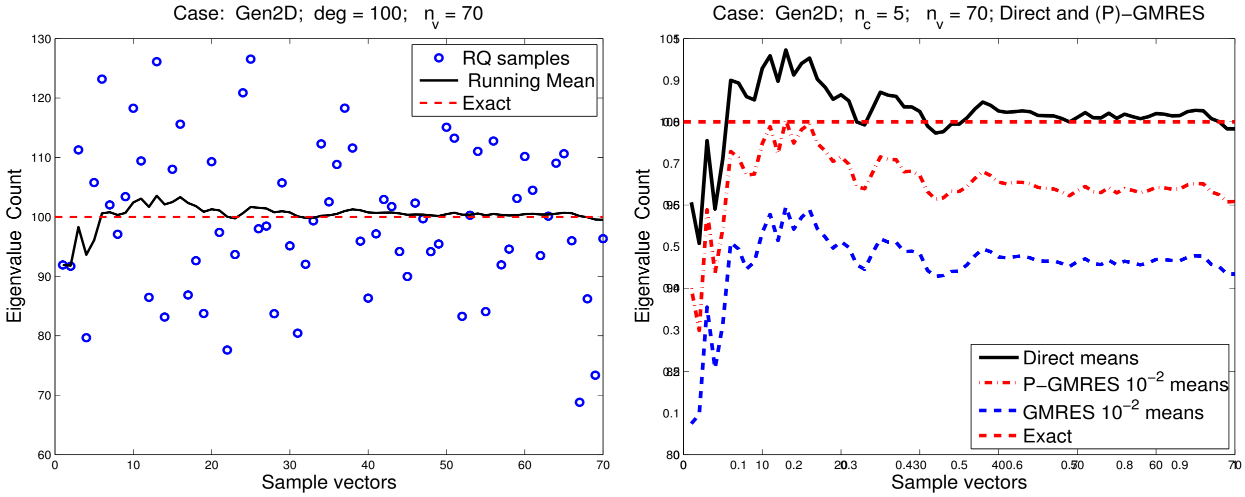


Figure 9: (Standard) Chebyshev vs rational approximation for the Gen2D matrix. The sequences of random vectors used in all the experiments are identical.

5.8 Rational approximation filtering for non-symmetric problems

Since the eigenvalues of any projector P, whether orthogonal or not, are equal to either zero or one, its trace is always equal to the number of its nonzero eigenvalues, which in the case under consideration is the number of eigenvalues located inside the contour integral. The stochastic estimator also works for non-symmetric matrices. Therefore, the whole technique extends to non-Hermitian case without any difference. The contour integration approach for computing the eigenpairs in a given region of the complex plane has also been successfully utilized within the framework of the FEAST solver [11].

Here we present some preliminary results on the applicability of the eigenvalue count estimate (11) for non-symmetric problems. In order to illustrate the accuracy of this approach, we select the complex symmetric matrix 'qc324' available from the Matrix Market [4] which is of size $n = 324$. Using a circle centered at (0,0) with radius 0.04, the exact (complex) eigenvalue count is 37. In order to estimate the trace $\mathtt{tr}(\mathtt{P})$ one generates complex random vectors $v_k, k = 1, .., n_v$ with entries ± 1 for both the real and imaginary parts. The imaginary part of the trace is expected to be very small, so the estimates are obtained using the real part of the trace in our simulation results. Figure 10 presents the estimates obtained using two different contour integrations. One notes that $n_c = 6$ (i.e. Gauss-3 for each half-circle) already provides some reasonable estimates. Further work would be needed to report a detailed study of the accuracy of the approach, but this preliminary result shows promise on the potential extension of the estimated eigenvalue counts to the complex plane.

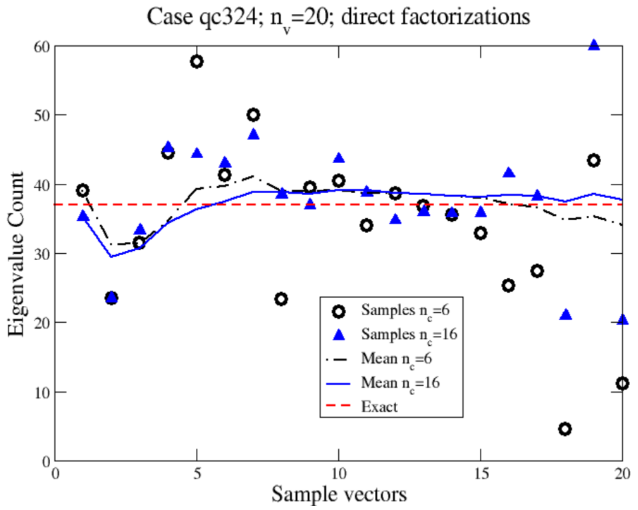


Figure 10: The rational approximation method at work for the qc324 matrix using a direct solver. The plots compare a run using 6 and 16 contours points for integration (obtained by placing respectively 3 and 8 Gauss contour points in the half-circle) and 20 sample vectors.

6 Conclusion

The methods presented in this paper rely on a compromise between accuracy and speed. If one is interested in an exact count, then clearly combining the Sylvester inertia theorem with some direct solution method may be the best option, although this may be too costly or even impractical in some situations. Otherwise, a stochastic estimate based on approximating the trace of the eigen-projector by exploiting a polynomial or rational function expansion, will

[4] http://math.nist.gov/MatrixMarket/

be sufficient. The rational approximation viewpoint may be perfectly suitable within the context of a package like FEAST since (approximate) factorizations will be needed at the outset anyway. The initialization of the package will begin by estimating the eigenvalue count in order to determine the proper subspace size to use. The additional cost of this step then remains relatively small and its use may lead to great savings. In other applications the polynomial approximation can give a good estimate at a relatively low cost.

Since the proposed methods are all based on an approximation of the spectral projector, they are subject to a slight bias if this approximation is not accurate enough. This bias may be exacerbated by the presence of clusters near the interval boundaries, since it is generally near these locations that the inaccuracies of the approximate projector are large. Finding reliable methods to reduce this bias remains an open issue that is worth investigating.

References

[1] C. Bekas, E. Kokiopoulou, and Y. Saad, *An estimator for the diagonal of a matrix*, Applied Numerical Mathematics, 57 (2007), pp. 1214 – 1229. Numerical Algorithms, Parallelism and Applications (2).

[2] T. A. Davis, *Direct methods for sparse linear systems*, SIAM, Philadelphia, PA, 2006.

[3] 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.

[4] E. Polizzi and P. Tang, *Subspace iteration with approximate spectral projection.* arXiv:1302.0432, 2013.

[5] FEAST *solver*, 2009-2013. http://www.ecs.umass.edu/∼polizzi/feast/.

[6] J. A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[7] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 3rd ed., 1996.

[8] M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Statist. Simula., 19 (1990), pp. 433–450.

[9] L. O. Jay, H. Kim, Y. Saad, and J. R. Chelikowsky, *Electronic structure calculations using plane wave codes without diagonlization*, Comput. Phys. Comm., 118 (1999), pp. 21–30.

[10] L. Krämer, E. D. Napoli, M. Galgon, B. Lang, and P. Bientinesi, *Dissecting the {FEAST} algorithm for generalized eigenproblems*, Journal of Computational and Applied Mathematics, 244 (2013), pp. 1 – 9.

[11] S. Laux, *Solving complex band structure problems with the feast eigenvalue algorithm*, Phys. Rev. B, 86 (2012), p. 075103.

[12] L. Lin, C. Yang, and Y. Saad, *Approximating spectral densities of large matrices*, Tech. Rep. ys-2013-1, Dept. Computer Science and Engineering, University of Minnesota, Minneapolis, MN, 2013. Submitted.

[13] L. Lopez and V. Simoncini, *Analysis of projection methods for rational function approximation to the matrix exponential*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 613–635.

[14] G. A. Parker, W. Zhu, Y. Huang, D. K. Hoffman, and D. J. Kouri, *Matrix pseudo-spectroscopy: iterative calculation of matrix eigenvalues and eigenvectors of large matrices using a polynomial expansion of the dirac delta function*, Comp. Phys. Comm., 96 (1996), pp. 27–35.

[15] E. Polizzi, *A density matrix-based algorithm for solving eigenvalue problems*, phys. rev. B, 79 (2009).

[16] E. Polizzi, *A high-performance numerical library for solving eigenvalue problems.* arXiv:1203.4031, 2013.

[17] H. Roder, R. Silver, D. Drabold, and J. Dong, *The kernel polynomial method for non-orthogonal electronic structure calculation of amorphous diamond*, Phys. Rev. B, 55 (1997), pp. 15382–15385.

[18] Y. Saad, *Filtered conjugate residual-type algorithms with applications*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 845–870.

[19] ——, *Numerical Methods for Large Eigenvalue Problems- classics edition*, SIAM, Philadelpha, PA, 2011.

[20] T. Sakurai and H. Sugiura, *A projection method for generalized eigenvalue problems using numerical integration*, Journal of Computational and Applied Mathematics, 159 (2003), pp. 119 – 128. ¡ce:title¿6th Japan-China Joint Seminar on Numerical Mathematics; In Search for the Frontier of Computational and Applied Mathematics towar d the 21st Century¡/ce:title¿.

[21] G. Schofield, J. R. Chelikowsky, and Y. Saad, *A spectrum slicing method for the kohn-sham problem*, Computer Physics Communications, 183 (2012), pp. 497–505.

[22] R. Silver, H. Roeder, A. Voter, and J. Kress, *Kernel polynomial approximations for densities of states and spectral functions*, Journal of Computational Physics, 124 (1996), pp. 115 – 130.

[23] J. M. Tang and Y. Saad, *A probing method for computing the diagonal of a matrix inverse*, Numerical Linear Algebra with Applications, 19 (2011), pp. 485–501.

[24] L. W. Wang, *Calculating the density of sates and optical-absorption spectra of large quantum systems by the plane-wave moments method*, Phys. Rev. B, 49 (1994), pp. 10154–10158.