A Quasi-Random Approach to Matrix Spectral Analysis*†

Michael Ben-Or[1] and Lior Eldar[‡2]

1 **Hebrew University, Jerusalem, Israel**
benor@cs.huji.ac.il
2 **MIT, Cambridge, USA**
leldar@mit.edu

Abstract

Inspired by quantum computing algorithms for Linear Algebra problems [6, 14] we study how simulation on a classical computer of this type of "Phase Estimation algorithms" performs when we apply it to the Eigen-Problem of Hermitian matrices. The result is a completely new, efficient and stable, parallel algorithm to compute an approximate spectral decomposition of any Hermitian matrix. The algorithm can be implemented by Boolean circuits in $O(\log^2 n)$ parallel time with a total cost of $O(n^{\omega+1})$ Boolean operations. This Boolean complexity matches the best known $O(\log^2 n)$ parallel time algorithms, but unlike those algorithms our algorithm is (logarithmically) stable, so it may lead to actual implementations, allowing fast parallel computation of eigenvectors and eigenvalues in practice.

Previous approaches to solve the Eigen-Problem generally use randomization to avoid bad conditions - as we do. Our algorithm makes further use of randomization in a completely new way, taking random powers of a unitary matrix to randomize the phases of its eigenvalues. Proving that a tiny Gaussian perturbation and a random polynomial power are sufficient to ensure almost pairwise independence of the phases (mod 2π) is the main technical contribution of this work. It relies on the theory of low-discrepancy or quasi-random sequences - a theory, which to the best of our knowledge, has not been connected thus far to linear algebra problems. Hence, we believe that further study of this new connection will lead to additional improvements.

1998 ACM Subject Classification G.1.3 Numerical Linear Algebra

Keywords and phrases Eigenvectors, Eigenvalues, low-discrepancy sequence

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.6

1 Introduction

1.1 General

The eigen-problem of Hermitian matrices is the problem of computing the eigenvalues and eigenvectors of a Hermitian matrix. This problem is ubiquitous in computer science and engineering, and because of its relatively high computational complexity imposes a high computational load on most modern information processing systems.

* The full version of this paper is available on line at https://arxiv.org/abs/1505.08126
† This research project was supported in part by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11), by the Israeli Science Foundation (ISF) research grant 1446/09, by an EU FP7 ERC grant (no.280157), and by the EU FP7-ICT project QALGO (FET-Proactive Scheme).
‡ LE is thankful to the Templeton Foundation for their support of this work.

Eigenvalues and eigenvectors of an input Hermitian matrix, even specified to finite precision, can be irrational numbers. Hence, when computing them, one inherently needs to approximate them. This gives rise to a host of problems: spectral decomposition algorithms are often hard to analyze rigorously, and turn out to be unstable, and difficult to parallelize.

Thus, given a matrix A, we are usually interested not in its exact eigenvalues and eigenvectors, which may be very hard to compute, (and possibly very long to describe once computed), but rather in an approximate decomposition:

▶ **Definition 1** (Approximate Spectral Decomposition - $\mathrm{ASD}(\mathrm{A}, \delta)$)**.** Let A be some $n \times n$ Hermitian matrix. An approximate spectral decomposition of A, with accuracy parameter $\delta = 1/\mathrm{poly}(n)$ is a set of vectors $\{v_i\}_{i=1}^n, \|v_i\| = 1$ such that there exists a complete set of eigenvectors $\{w_i\}_{i=1}^n$ of a matrix A', $\|A' - A\| \leq \delta$ that satisfy:

$$\forall i \quad \|v_i - w_i\| \leq \delta.$$

For a general $n \times n$ matrix A one can consider the Hermitian matrix $A^H A$, in which case $\mathrm{ASD}(\mathrm{A^H A}, \delta)$ is an approximation of the *singular vectors* (and singular values) of A.

We note that the definition of ASD then corresponds to a "smooth analysis" of matrices: namely given input A, we do not find a spectral decomposition of A, but rather the decomposition of a matrix A', such that $\|A - A'\| \leq \delta$. We also point out, that the definition of ASD holds just as well in the case of nearly degenerate matrices: we do not require a one-to-one correspondence with the eigenvectors of A, which can be extremely hard to achieve, but rather to find some set of approximate eigenvectors, such that the corresponding weighted sum of rank-1 projections form an approximation of A.

When one considers an algorithm $\mathcal{A}$ for the ASD problem, one can examine its *arithmetic* complexity or *boolean* complexity. The arithmetic complexity is the minimal size arithmetic circuit C (namely each node computes addition, multiplication or division to unbounded accuracy) that implements $\mathcal{A}$, whereas the boolean complexity counts the number of boolean AND/OR gates of fan-in 2 required to implement $\mathcal{A}$.

Given the definition above, and following Demmel et al. [4] we consider an algorithm $\mathcal{A}$ to be log-stable (or stable for short), if there exists an arithmetic circuit C that implements $\mathcal{A}$ on $n \times n$ matrices, and a number $t = O(\log(n))$, such that each arithmetic computation in C uses at most t bits of precision, and the output of the circuit deviates from the output of the arithmetic circuit by at most $1/\mathrm{poly}(n)$. We note that when an algorithm is *stable* then its boolean complexity is equal to its arithmetic complexity up to a factor $O(\log(n))$. If, however, an algorithm is *unstable* then its boolean complexity could be larger by a factor of up to n. In the study of practical numerical linear algebra algorithms, one usually identifies algorithms that are stable with "practical", and algorithms that are not stable to be impractical. This usually, because the computing machines are restricted to representing numbers with a number of bits that is a small fraction of the size of the input.

In terms of parallelism, we will refer to the complexity class $\mathrm{NC^{(k)}}$ (see Definition 8) which is the set of all computational problems that can be solved by uniform Boolean circuits of size $\mathrm{poly}(n)$ in time $O(\log^k(n))$. Often, we will refer to the class $\mathrm{RNC^{(k)}}$, in which the parallel $\mathrm{NC^{(k)}}$ circuit is also allowed to accept uniform random bits. One would like an ASD algorithm to have minimal arithmetic or boolean complexity, and minimal parallel time. Ideally, one would also like this algorithm to be stable.

1.2 Main Contribution

Inspired by recent quantum computing algorithms [6, 14], we introduce a new perspective on the problem of computing the ASD that is based on low-discrepancy sequences. Roughly

speaking, low-discrepancy sequences are deterministic sequences which appear to be random, because they "visit" each small sub-cube the same number of times that a completely random sequence would, up to a small additive error.

▶ **Definition 2** (Multi-dimensional Discrepancy)**.** For integer s, put $I^s = [0,1)^s$. Given a sequence $x = (x_n)_{n=1}^N$, with $x_n \in I^s$ the discrepancy $D_N(x)$ is defined as:

$$D_N(x) = \sup_{B \in \mathcal{B}} \left\{ \left| \frac{1}{N} \sum_{n=1}^{N} \chi_B(x_n) - \mathrm{vol}(B) \right| \right\},$$

where $\chi_B(x_n)$ is an indicator function which is 1 if $x_n \in B$ and $\mathcal{B}$ is the set of all s-products of intervals $\prod_{i=1}^s [u_i, v_i]$, with $[u_i, v_i](\mathrm{mod}1) \subseteq [0,1)$.

We recast the ASD problem as a question about the discrepancy of a certain sequence related to the input matrix. Specifically, given a Hermitian matrix A with n unique eigenvalues $\{\lambda_i\}_{i\in[n]}$ the central object of interest is the sequence comprised of n-dimensional vectors of eigenvalue residuals:

$$S(A) = \left(\{\lambda_1 \cdot 1\}, \ldots, \{\lambda_n \cdot 1\}\right), \left(\{\lambda_1 \cdot 2\}, \ldots, \{\lambda_n \cdot 2\}\right), \ldots \left(\{\lambda_1 \cdot M\}, \ldots, \{\lambda_n \cdot M\}\right),$$

where $\{x\}$ is the fractional part of $x \in \mathbb{R}$, and $M = \mathrm{poly}(n)$ is some large integer. $S(A)$ is hence a sequence of length M in $[0,1)^n$. We would like $S(A)$ to have as small discrepancy as possible. Hence, in sharp contrast to previous algorithms, instead of the computational effort being concentrated on revealing "structure" in the matrix, our algorithm is actually focused on producing random-behaving dynamics.

The main application of our approach presented in this paper is a new stable and parallel algorithm for computing the ASD of any Hermitian matrix. We assume w.l.o.g. that the input matrix is positive-semidefinite (otherwise it can be scaled and shifted by appropriate multiple of identity) and claim:

▶ **Theorem 3.** *Let A be some $n \times n$ Hermitian matrix such that $0 \preceq A \preceq 0.9I$. Let $\delta = 1/\mathrm{poly}(n)$. Then $\mathrm{ASD}(\mathrm{A}, \delta) \in \mathrm{RNC}^{(2)}$, with circuit size $\tilde{O}(n^{\omega+1})$. The algorithm is* log*-stable.*

The boolean complexity of our algorithm is $O(n^{\omega+1})$. If however, one is interested in sampling a uniformly random eigenvector, it can be achieved in complexity $O(n^{\omega})$. [1]

1.3 Prior Art

There are numerous algorithms for computing the ASD of a matrix, relying most prominently on the QR decomposition [15]. For specific types of matrices, like tridiagonal matrices much faster algorithms are known [11], but here we consider the most general Hermitian case. We summarize the state of the art algorithms for this problems in terms of their complexity (boolean / arithmetic, serial / parallel) and compare them to our own:

[1] ω signifies the infimum over all constants c such that one can multiply two matrices in at most n^c arithmetic operations, and $O(\log(n))$ time.

	Arithmetic Complexity	Boolean Complexity	Parallel Time	Log-Stable	Comments
Csanky [7]	$\tilde{O}(n^{\omega+1})$	$\tilde{O}(n^{\omega+2})$	$\log^2(n)$	NO	
Demmel et al. [4]	$\tilde{O}(n^{\omega})$	$\tilde{O}(n^{\omega})(*)$	N/A	YES	$*$ Conjectured for a variant of the algorithm.
Bini et al., Reif [3, 11]	$\tilde{O}(n^{\omega})$	$\tilde{O}(n^{\omega+1})$	$O(\log^2(n))$	NO	Working with $\Omega(n)$ bit Integers
New	$\tilde{O}(n^{\omega+1})$	$\tilde{O}(n^{\omega+1})$	$\log^2(n)$	YES	

Comparing our algorithm to the best known $\mathrm{NC}^{(2)}$ algorithms, it is more efficient by a factor of n compared with Csanky's algorithm [7]. Notably, our algorithm is completely disjoint from Csanky's techniques - which rely on computing explicitly high powers of the input matrix, and computes the characteristic polynomial of the matrix using the Newton identities on the traces of those powers. This is an inherently unstable algorithm as it finds the eigenvalues by approximating the roots of the characteristic polynomial and small perturbation to the coefficients of the polynomial may lead to large deviations of the roots.

The algorithms of Demmel et al., Bini et al. and Reif, rely on efficient implementation of variants of the QR algorithm. Our asymptotic bounds are worse then Demmel et al. in terms of total arithmetic/boolean complexity, though we conjecture that this is an artifact of our proof strategy, and not an inherent problem (see the section on open problems), and in fact, we conjecture that a certain variant of the algorithm could probably achieve a boolean complexity of $O(n^{\omega})$. We note that the QR algorithm is not known to be parallelizable in a stable way, and hence the fast parallel algorithms of Bini et al. and Reif are not stable and probably impractical. In fact the QR decomposition has been shown, for standard implementations like the Given's or Householder method, to be P-complete [8] assuming the real-RAM model. Thus, it is unlikely to be stably-parallelizable unless $\mathrm{P} = \mathrm{NC}$. [2]

Thus, to the best our knowledge, our algorithm is the first parallel algorithm for the ASD of general Hermitian matrices that is both parallel and stable. In particular it achieves the smallest bit-complexity of any $\mathrm{RNC}^{(2)}$ algorithm to date. We conjecture that our approach may present a practical and parallel alternative to computing the ASD.

1.4 Overview of the Algorithm

To compute the ASD of a given matrix A, we first consider a similar problem of sampling uniformly an approximate eigenvector of A, where the eigenvalues of A are assumed to be well-separated. Clearly, if one can sample from this distribution in RNC^2, then by the coupon collector's bound concatenating $O(n\log(n))$ many parallel copies of this routine, one can sample all eigenvectors quickly with high probability. To do this, we require a definition of a Hermitian matrix that is δ-separated:

▶ **Definition 4** (δ-separated)**.** Let A be an $n \times n$ PSD matrix with eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_n \geq 0$. We say that A is δ-separated if $\lambda_j - \lambda_{j+1} \geq \delta$ for all $j < n$, and $\lambda_1 \leq 1/(2\pi) - \delta$.

Next, we introduce the notion of a separating integer w.r.t. a sequence of real numbers:

[2] We point out that the algorithm of Reif [11] achieves a QR factorization in parallel time $O(\log^2(n))$ in the arithmetic model, thus showing that QR is indeed parallelizable, but it relies on computations modulo large integers and therefore not stable and not practical.

▶ **Definition 5** (Separating Integer). Let $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n) \in [0,1)^n$. For $\alpha > 4$ define $B_{in} \subseteq B_{out} \subseteq [0,1)$ as:

$$B_{out} = [-1/(4n), 1/(4n)](\text{mod}1) \quad \text{and} \quad B_{in}(\alpha) = [-1/(\alpha n), 1/(\alpha n)](\text{mod } 1),$$

A positive integer m is said to separate the k-th element of $\bar{\lambda}$ w.r.t. B_{in}, B_{out} if it satisfies:

- $\{m\lambda_k\} \in B_{in}(\alpha)$
- $\forall j \neq k \;\; \{m\lambda_j\} \notin B_{out}$

and finally define the notion of a separating integer w.r.t. a δ-separated matrix.

▶ **Definition 6.** Let A be a δ-separated matrix with eigenvalues $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n)$. A positive integer m is said to separate k in A w.r.t. B_{in}, B_{out}, if m separates the k-th element of $\bar{\lambda}$ (namely, the k-th eigenvalue of A) w.r.t. B_{in}, B_{out}.

Following is a sketch of the main sampling routine. For complete details see Section 5. The routine accepts a separating integer m of the i-th eigenvalue of a δ-separated matrix A, a precision parameter δ and returns a δ approximation of the i-th eigenvector of A:

Algorithm 1 Filter(A, m, δ)

Input: $n \times n$ Hermitian matrix $A \succeq 0$, integer m, $\delta = 1/\text{poly}(n)$. A is δ-separated.

1. **Compute parameters:**
   $$p = 2n^2\lceil \ln(1/\delta) \rceil, \zeta = \delta^2/(2pm).$$
2. **Sample random unit vector:**
   Sample a standard complex Gaussian vector v, set $w_0 = v/\|v\|$.
3. **Approximate matrix exponent:**
   Compute a ζ Taylor approximation of $e^{2\pi i A}$, denoted by $\tilde{U}$.
4. **Raise to power:**
   Compute $\tilde{U}^m$ by repeated squaring.
5. **Generate matrix polynomial:**
   Compute $B = \left(\frac{I+\tilde{U}^m}{2}\right)^p$ by repeated squaring.
6. **Filter:**
   Compute $w = \frac{B \cdot w_0}{\|B \cdot w_0\|}$.
7. **Decide:**
   Set $z = A \cdot w$, $i_0 = \arg\max_{i \in [n]} |w_i|$ and compute $c = z_{i_0}/w_{i_0}$. If
   $$\|A \cdot w - c \cdot w\| \leq 3\delta\sqrt{n}$$
   return w, and otherwise reject.

In words - the algorithm samples a random vector and then multiplies it essentially by the matrix $B = ((I + e^{2\pi i A m})/2)^p$. After this "filtering" step, it evaluates whether or not the resulting vector is close to being an eigenvector of A, and keeps this vector if it is. To understand the behavior of the algorithm, it is insightful to consider the behavior in the eigenbasis of A.

$$w = \sum_i \alpha_i w_i,$$

where $\{w_i\}_{i\in[n]}$ is an orthonormal basis for A corresponding to eigenvalues $\{\lambda_i\}_{i\in[n]}$. If $\{m\lambda_i\}$, i.e. - the fractional part of $m\lambda_i$, is very close to 0 (i.e. inside B_{in}) and $\{m\lambda_j\}$ is $\sim 2\ln n/p$ far from 0 (i.e. outside B_{out}) for all $j \neq i$, then after multiplication by B and normalization, all eigenvectors w_j for $j \neq i$ are attenuated by factor $1/n^2$ relative to w_i, and hence the resulting vector is $1/n$ close to an *eigenvector* of λ_i.

Hence, a sufficient condition on the number m that would imply that $w = Filter(A, m, \delta)$ is an approximation of the i-th eigenvector is the following property: $\{m\lambda_i\}$ is very close to 0, and for all $j \neq i$ $\{m\lambda_j\}$ is bounded away from 0. This corresponds to the fact that m *separates* i in A, as assumed.

So to sample uniformly an approximate eigenvector, we would like to call $Filter(A, m, \delta)$ for $m \sim U[M]$ for $M = \text{poly}(n)$ and prove that m separates i where $i \sim U[n]$. The main observation here, is that this property holds if the sequence of residuals of integer multiples of the eigenvalues $S(A)$ defined above has the aforementioned *low discrepancy* property.

Most of the work in this study is devoted to achieving this property. Computationally, we achieve low-discrepancy of $S(A)$ simply by additive Gaussian perturbation prior to calling the sampling routine. We show that if we perturb a matrix using a Gaussian matrix $\mathcal{E}$ of variance $1/\text{poly}(n)$, then $S(A + \mathcal{E})$ has discrepancy which is $1/\text{poly}(n)$. Showing this is non-trivial because arbitrary vectors of eigenvalues $\lambda_1, \ldots, \lambda_n$ do not generate low-discrepancy sequences in general, and on the other hand we are also severely limited in our ability to perturb the eigenvalues without deviating too much from the original matrix. This is the subject of our main technical theorem 34, which may be of independent interest:

▶ **Theorem** (Informal). *Let A be an $n \times n$ Hermitian matrix, and $\mathcal{E}$ be a standard Gaussian matrix. For any $a > 0, b > 0$ there exists $M = M(a, b) = \text{poly}(n)$ such that w.p. at least $1 - n^{-b}$ the sequence of residuals of eigenvalue multiples of $A + n^{-a} \cdot \mathcal{E}$ of length M has discrepancy at most n^{-b}.*

Perturbing the input matrix has the additional benefit of making sure that A has a exactly n unique eigenvalues with high probability. This follows from a breakthrough theorem by Nguyen, Tao and Vu [9] which has provided a resolution of this long-standing open problem, which was considered unproven folklore until that point. This theorem allows us to handle general Hermitian matrices without extra conditions on the conditioning number of A or its eigenvalue spacing.

1.4.1 Comparison to the power method / QR algorithm

A natural benchmark by which to test the novelty of the proposed algorithm is the iterative power-method for computing the eigenvalues of a Hermitian matrix. In this method, one starts from some random vector b_0, and at each iteration k sets:

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|}.$$

Both the power method and our proposed scheme are similar in the sense that they attempt to extract the eigenvectors of the input matrix directly. Also, if two eigenvalues are ε-close in magnitude, for some $\varepsilon > 0$, then they require essentially the same exponent of A in the power method, and of e^{iA} in our scheme to distinguish between them. However, the similarity stops here. We maintain, that the power method is both conceptually different, and for general Hermitian matrices performs much worse, in terms of running time, compared with our proposed algorithm.

Conceptually, in the power method, we seek to leverage the difference in *magnitude* between adjacent eigenvalues in order to extract the eigenvectors. On the other hand, in our proposed scheme we recast the problem on the unit sphere $S^{(1)}$, where we are interested in the spacing of the residuals of integer multiples of the eigenvalues. Worded differently, our setting exploits the additive group structure of the eigenvalues modulo 1, whereas the power method distinguishes between them multiplicatively.

In the additive group setting, the advantage is that we can consider the discrepancy of the sequence of residuals, and analyze how quickly these residuals mimic a completely independent random distribution. Furthermore, in the additive setting there is inherent symmetry between the eigenvalues, as no eigenvalue is more likely to be sampled than another. This allows for a natural parallelization of the algorithm to extract simultaneously approximation of all eigenvectors.

The well-known QR algorithm for eigendecomposition [5] is the de-facto standard for computing the ASD, and is considered by some as a parallel version of the power-method. That algorithm applies an iterated sequence of of QR decompositions: At each step k we compute (where $A_1 = A$ - the input matrix)

$$A_k = Q_k R_k,$$

and then set

$$A_{k+1} = R_k Q_k.$$

The algorithm runs in time $\tilde{O}(n^3)$, by applying several pre-processing steps [5], and the fast variant of Demmel et al. in time $O(n^\omega)$. However, as stated above, the QR decomposition which is at the core of these methods is not known to be stably parallel.

1.5 Open Questions

We outline several open questions that may be interesting to research following this work:

1. Is it possible to attain a serial run-time of $O(n^\omega)$ for this algorithm? We conjecture that this is possible based on numerical evidence for a variant of this algorithm, yet we do not have a proof of this fact.
2. What other linear-algebra algorithms can be designed using our methods ? We would like these algorithms to improve on previous algorithms in either the stability, boolean complexity, parallel run-time, or all these parameters simultaneously.
3. Could one reduce the number of random bits required by the algorithm? Currently - we show that using $\tilde{O}(n^2)$ random bits - i.e. applying additive Gaussian perturbation results in a matrix whose eigenvalues seed a low-discrepancy sequence. However, can one do away with only $\tilde{O}(n)$ random bits - by applying a tri-diagonal perturbation to the matrix?

2 Preliminaries

2.1 Notation

A random variable x distributed according to distribution $\mathcal{D}$ is denoted by $x \sim \mathcal{D}$. We will use the letter D to denote the *discrepancy* of a sequence, and the calligraphy letter $\mathcal{D}$ to denote a distribution. For a matrix X, $\|X\|$ signifies the operator norm of X. For a set S, $U[S]$ is the uniform distribution on S. For integer $M > 0$ the set $[M]$ is the set of integers

$\{0, 1, \ldots, M-1\}$. For real number x, $\{x\}$ denotes the fractional part of x: $\{x\} = x - \lfloor x \rfloor$. For real number x, $\lceil x \rfloor \in [-1/2, 1/2)$ denotes the rounding error of x - i.e. $\min\{x - \lceil x \rceil, x - \lfloor x \rfloor\}$. $\mathbb{N}, \mathbb{Z}, \mathbb{C}$ signify the natural, integer, and complex numbers, respectively. For a matrix A, A^H is the Hermitian conjugate-transpose of A. For number $n > 0$ $\ln n$ denotes the natural logarithm, and $\log n$ denotes the binary logarithm. $\mu(\eta, \sigma^2)$ is the Gaussian measure with mean η and variance σ^2. An n-dimensional vector is σ-normal if its components are i.i.d. $\mu(0, \sigma^2)$. $U(n)$ is the set of $n \times n$ unitary matrices. For a Hermitian $n \times n$ matrix A, with eigenvalues $\{\lambda_i\}_{i=1}^n$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ $\mathcal{L}(A) = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$ denotes the vector of sorted eigenvalues of A. For a measurable subset $S \subseteq \mathbb{R}^n$ $\mathrm{vol}(S)$ denotes the volume of S. $\emptyset$ is the empty set. GUE is the global unitary ensemble of random matrices: these are Hermitian matrices whose upper-triangular entries are independently sampled as $\mu(0, 1)$.

2.2 Definitions

2.2.1 Complexity

▶ **Definition 7.** Let ω denote the infimum over all t such that any two $n \times n$ matrices can be multiplied using a number of products at most n^t, and time $O(\log(n))$.

The current best upper-bound on ω is 2.372 due to Williams [16].

▶ **Definition 8** (Class NC)**.** The class $\mathrm{NC}^{(k)}$ is the set of problems computed by uniform boolean circuits, with a polynomial number of gates, and depth at most $O(\log^k n)$.

Additionally, we will require the following fact:

▶ **Fact 9** ([1])**.** *There exists an algorithm for sorting n numbers in time $O(\log(n))$, using n processors.*

▶ **Definition 10** (Class RNC)**.** The class $\mathrm{RNC}^{(k)}$ is the set of problems that can be computed by uniform boolean circuits, with a polynomial number of gates, accepting a polynomial number of random bits, and depth at most $O(\log^k n)$.

For simplicity, we shall assume in this work that RNC circuits are allowed to accept t-bit numbers, sampled from a suitably truncated Gaussian distribution, and discretized to t-bits of precision.

2.2.2 Stable Computation

Following Demmel et al. [4] we define the notion of log-stability as one where truncating each binary arithmetic operation to $O(\log(n))$ bits of precision doesn't change the result by much:

▶ **Definition 11** ((t, δ)-stable randomized computation)**.** Let C denote a randomized arithmetic circuit, and $\mathcal{D}$ be its output distribution supported on $\mathbb{R}^n$. Let D denote the discretization of C to t bits as follows: each infinite-precision arithmetic operation is followed by rounding to t bits. Let $\mathcal{D}'$ denote the output distribution of D. C is said to be (t, δ)-stable if

$$\forall x \quad \exists y, \quad \mathcal{D}(x) = \mathcal{D}'(y) \text{ and } \|x - y\| \leq \delta.$$

▶ **Definition 12** (Log-stable computation)**.** Let C be a randomized arithmetic circuit that accepts n input numbers. C is said to be log-stable if for any $\delta = 1/\mathrm{poly}(n)$ it is (t, δ)-stable for some $t = O(\log(1/\delta))$.

3 Additive Perturbation

Matrix perturbation is a well-developed theory [13, 5] examining the behavior of eigen-values and eigen-vectors under additive perturbation, usually much smaller compared to the norm of the original matrix. While general eigenvalue problems are usually unstable against perturbation, for Hermitian matrices the situation is much better: the Bauer-Fike theorem [2] states that the perturbed eigenvalues can only deviate from the original eigenvalues by an amount corresponding to the relative strength of the perturbation. This holds regardless of whether the perturbation itself is Hermitian.

In particular, when the perturbed matrix A is δ-separated and the perturbation itself is Hermitian (GUE, for example) one can compute an explicit estimate for the behavior of the perturbed eigenvalues. We use here a quantitative estimate by [12]:

▶ **Fact 13** (Stability of well-separated eigenvalues under perturbation). *Let A be a δ-separated $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_n$, and corresponding orthonormal basis $\{v_i\}_{\in[n]}$. Let $\mathcal{E}$ be an additive perturbation of A satisfying $|\mathcal{E}_{i,j}| \leq \varepsilon$ for all i, j. Let $\tilde{\lambda}_i$ denote the i-th eigenvalue of $A + \mathcal{E}$. There exists a constant $c > 0$ satisfying:*

$$\forall i \in [n] \quad \tilde{\lambda}_i = \lambda_i + v_i^H \mathcal{E} v_i + \zeta_i, \quad |\zeta_i| \leq c\varepsilon^2/\delta.$$

In fact, if the perturbation $\mathcal{E}$ is GUE a stronger characterization is readily available:

▶ **Corollary 14.** *Let A be a δ-separated $n \times n$ Hermitian matrix with eigenvalues $\{\lambda_i\}_{i\in[n]}$, and corresponding orthonormal basis $\{v_i\}_{\in[n]}$. Let $\mathcal{E}$ be GUE. There exists $c > 0$ independent of n such that the eigenvalues $\{\lambda'_i\}_{i\in[n]}$ of the perturbed matrix $A' = A + \varepsilon \cdot \mathcal{E}$ are distributed as follows: they are sampled from $\mu(\lambda_i, \varepsilon^2)$, and added a number ζ_i satisfying w.p. $1 - 2^{-\Omega(n)}$:*

$$|\zeta_i| \leq cn \cdot \varepsilon^2/\delta$$

Proof. By Fact 13 the eigenvalues λ'_i behave as

$$\lambda'_i = \lambda_i + v_i^H \mathcal{E} v_i + \zeta_i, \quad |\zeta_i| \leq c\varepsilon^2 \max_{i,j} |\mathcal{E}_{i,j}|^2/\delta,$$

for some constant $c > 0$. The random matrix $\mathcal{E}$ is invariant under unitary conjugation so in particular, for the unitary matrix V whose columns are the v_i's we have:

$$V^H \mathcal{E} V \sim \mathcal{E}$$

which implies

$$\lambda'_i = \lambda_i + \mathcal{E}_{i,i} + \zeta_i,$$

where

$$|\zeta_i| \leq c \max_{i,j} \left|(V^H \mathcal{E} V)_{i,j}\right|^2/\delta \sim c\varepsilon^2 \max_{i,j} |\mathcal{E}_{i,j}|^2/\delta.$$

The standard Gaussian satisfies:

$$\mathsf{P}_\mu\left(|x| \geq 4\sqrt{n}\right) \leq 2^{-2n}.$$

Thus, by the union bound we have that $|\mathcal{E}_{i,j}| \leq 4\sqrt{n}$ for all i, j w.p. at least $1 - 2^{-n}$. Hence, w.p. at least $1 - 2^{-n}$ we have:

$$\forall i \in [n] \quad |\zeta_i| \leq c \max_{i,j} |\mathcal{E}_{i,j}|^2/\delta \leq 16cn \cdot \varepsilon^2/\delta, \qquad \blacktriangleleft$$

Our interest in additive perturbation, however, is not confined just to "stability" arguments. In fact, our main reason for using perturbation is to cause a scattering of the eigenvalues. The first step of our algorithm in fact applies additive perturbation to provide a minimal spacing between eigenvalues. Recently Nguyen et al. [9] have shown that applying additive perturbation to any Hermitian matrix using a the well-known Wigner ensemble, an ensemble of random matrices that generalize GUE, in fact causes the eigenvalues of the perturbed matrix to achieve a minimal inverse polynomial separation. We state their result:

▶ **Lemma 15** ([9], Theorem 2.6. Minimal eigenvalue spacing)**.** *Let $M_n = F_n + \varepsilon \cdot X_n$, where F_n is a real symmetric matrix, $\|F_n\|_2 \leq 1$, $\varepsilon = n^{-\gamma}$ for some constant $\gamma > 0$, and X_n is GUE - namely a random Hermitian matrix (see Section 2). Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ denote the eigenvalues of M_n, and put $\alpha_i = \lambda_i - \lambda_{i+1}$ for all $i < n$. Then for any fixed $A > 0$ there exists $B = B(A, \gamma) > 0$, such that*

$$\max_{1 \leq i < n} \mathsf{P}\left(\alpha_i \leq n^{-B}\right) = \mathrm{O}(n^{-A}).$$

In particular [3] *for any $A > 0$ there exists $B > 0$ such that $\mathsf{P}\left(\min_{1 \leq i < n} \alpha_i \geq n^{-B}\right) = 1 - \mathrm{O}(n^{-A})$.*

Using the lemma above we define the number $B^* = B^*(\delta)$ as follows:

▶ **Definition 16.** For any $\delta = 1/\mathrm{poly}(n)$, let $B^*(\delta)$ denote the smallest number $B > 0$ such that for every F_n the matrix $M_n = F_n + \delta X_n$ satisfies:

$$\mathsf{P}\left(\min_{1 \leq i < n} \alpha_i \geq n^{-B}\right) \geq 0.99$$

4 Low-Discrepancy Sequences

4.1 Basic Introduction

Low discrepancy sequences (or "quasi-random" sequences) are a powerful tool in random sampling methods. Roughly speaking, these are deterministic sequences that visit any measurable subset B a number of times that is roughly proportional to the volume of B, up to some small additive error, called the discrepancy. See definition 2.

The definition of discrepancy naturally admits an interpretation in terms of probability:

▶ **Definition 17** (Discrepancy of a random variable)**.** Let x be a random variable on $[0,1)^s$. We define the discrepancy of x, $D(x)$ as follows:

$$D(x) = \max_{S \in \mathcal{B}} \left|\mathsf{P}_z(z \in S) - \mathrm{vol}(S)\right|.$$

By definition, if x is a sequence of length N of discrepancy $D_N(x)$, and z is a uniformly random element from x, then $D(z) = D_N(x)$.

Low-discrepancy sequences have much in common with random sampling, or the Monte-Carlo method, in the sense that they visit each cube a number of time that is roughly proportional to its volume, up to a small additive error. Yet, contrary to the Monte-Carlo method, such sequences are *not* random, but only appear to be random in the sense above.

[3] applying the union bound over all eigenvalues

There are deterministic s-dimensional sequences $x = \{x_i\}_{i=1}^N$ with discrepancy as low as

$$D_N(x) \leq C \cdot \frac{\log^s N}{N},$$

and matching lower-bounds (up to constant factors) on the smallest possible discrepancy are known for $s = 1$ [10]. Hence, usually one considers low-discrepancy sequences that are very long (N) compared to the dimension (s). In particular, in this work we will focus on attaining low-discrepancy sequences for dimension $s = 2$. 2-dimensional low-discrepancy sequences can be viewed as an approximation to pairwise independent uniform random variables on the interval $[0, 1)$. This property will be crucial in proving that we are able to isolate and "filter-out" single eigenvectors, and do so in a way that does not favor any particular eigenvector (see for example Lemma 36).

We mention, in passing, that the discrepancy upper-bound decays asymptotically almost as $O(1/N)$ (assuming small dimension s) whereas for a sequence $x = \{x_n\}_n$ where the x_n's are uniform independent samples (Monte-Carlo method) the discrepancy typically decays more slowly, behaving as $O(1/\sqrt{N})$ - and hence quasi-random sequences are often preferred as a method of numerical integration.

4.2 Some basic facts

We require a Lemma [2.5] due to Niederreiter [10].

▶ **Lemma 18** ([10]. Small point-wise distance implies similar discrepancy)**.** *Let $x_1, \ldots, x_N$, $y_1, \ldots, y_N$ denote two s-dimensional sequences for which $|x_{n,i} - y_{n,i}| \leq \varepsilon$, for all $n \in [N], i \in [s]$. Then the discrepancies of these sequences are related by:*

$$|D_N(x_1, \ldots, x_N) - D_N(y_1, \ldots, y_N)| \leq s \cdot \varepsilon. \tag{1}$$

We prove an additional fact:

▶ **Fact 19** (Monotonicity of discrepancy under addition of independent random variables)**.** *Let x be a random variable on $[0, 1)^s$ of discrepancy at most $D(x)$, and let y denote the random variable*

$$y = x + z (mod 1),$$

where z supported on $[0, 1)^s$ is a random variable independent of x. Then $D(y) \leq D(x)$.

The proof appears in the full version of the paper.

4.3 The Good Seed Problem

We will be interested in sequences $x = \{x_n\}_{n=1}^N$ where each x_n is an s-dimensional vector comprised of residuals of numbers as follows:

$$x_n = \{g \cdot n\},$$

where $g \in [0, 1)^s$ is some s-dimensional vector, called the *seed* of the sequence. Specifically, in our context, the vector g will represent the vector of eigenvalues of an $n \times n$ Hermitian matrix A whose spectrum we would like to analyze. Since it is unreasonable to assume that the input matrix has a spectrum that is a good seed, we will find a perturbation of the matrix $A' = A + \mathcal{E}$ such that $g' = \mathcal{L}(A')$ has a corresponding sequence, defined as above, with low-discrepancy.

Niederreiter has shown [10] that if g is sampled uniformly on $[N]^s$ then it is a good seed with high probability:

▶ **Lemma 20.** *Let s, N be an integers and $g \sim U([N]^s)$, and let $x = \{x_n\}_n$ denote the sequence whose n-th element is given by: $x_n = \{gn/N\}$. Then*

$$\mathsf{P}\left(D_N(x) \leq \frac{\log^s N}{N}\right) \geq 1 - 1/N.$$

For our application we require that $N = \text{poly}(n)$, and $s = 2$, in which case the above discrepancy is sufficiently low for our purposes. Yet, since it requires the normalized seed g/N to be essentially uniform on $[0,1)^n$, it implies that the corresponding matrix perturbation $\mathcal{E}$ added to A must be very strong - thereby loosing all connection to the input matrix.

4.4 Finding Reasonably-Good Seeds Locally

To bridge the gap between weak-perturbation and low-discrepancy we show a new lemma, which may be of independent interest: it allows to trade-off the extent to which g is random, and the discrepancy of the sequence generated by g. Specifically, we will show that if g/N is uniform on *cubes* of much smaller side-length, i.e. at least $1/\sqrt{N}$, then the resulting sequence has discrepancy $O(\log^s N/\sqrt{N})$. This is the subject of the following lemma:

▶ **Lemma 21.** *We are given integer N, with prime divisor M and an integer s. Let $g = (g_1, \ldots, g_s) \in N^s$, such that each coordinate g_i is independently chosen uniformly on some interval $I_i \subseteq [N]$ of size M. Let $x = x(g) = \{x_n\}_{n=1}^N$ be the following s-dimensional sequence of length N corresponding to residuals of g:*

$$x_n = \left\{\frac{g \cdot n}{N}\right\}.$$

Then $\mathsf{P}_g\left(D_N(x) \leq 2\log^s(M)/\sqrt{M}\right) \geq 1 - 1/\sqrt{M}$.

The proof appears in the full version of the paper.

4.5 Low-Discrepancy from Gaussian vectors

In the previous section we showed that sampling a vector of integers uniformly from an s-dimensional cube formed by the s-th fold product of an interval $M \subseteq [N]$ yields w.h.p. a sequences of discrepancy at most $1/\sqrt{M}$. In this section we adapt these theorems about good seeds for low-discrepancy sequences to the Gaussian measure: we show that sampling a vector $g = (g_1, \ldots, g_s)$ according to the Gaussian measure (e.g. "normal vector") with variance N^{-a} yields w.h.p. a sequence of discrepancy at most N^{-b} for some positive constants a, b. The proof of this is rather technical, and hinges on an approximation of the Gaussian measure of variance σ^2 by a convex combination of uniform distributions on intervals of size $\sigma/\text{poly}(N)$.

▶ **Theorem 22** (Approximating a Gaussian by a convex sum of uniform distributions)**.** *Let $g = (g_1, \ldots, g_n)$ be a vector $g \in \mathbb{R}^n$ sampled from the standard Gaussian measure. Then g is a convex combination of two distributions $\mathcal{D}_U, \mathcal{D}_V$ as follows: $(1-p)\mathcal{D}_U + p \cdot \mathcal{D}_V$, where $\mathcal{D}_U$ is the n-fold distribution of independent variables $z_1, \ldots, z_n$, and $p \leq 2n^2/m$. Each z_i is itself a convex combination of $m \geq 2n^2$ i.i.d. variables $\{w_j\}_{j=1}^m$, with $w_j \sim U[I_j]$, where I_j is some interval of the real line of size $|I_j| = 1/m$.*

The proof of this theorem is somewhat technical and appears in the full version of the paper. We now define a vector to be "almost" normal - in the sense that it is a small perturbation of a normal vector:

▶ **Definition 23** ((σ,ε)-normal vector)**.** A random vector v is (σ,ε)-normal if it is sampled as a σ-normal vector x to which we add a vector $e = e(x)$ of length at most $\sigma\varepsilon$.

We now state our main lemma of this section - that almost normal vectors yield seeds for low-discrepancy sequences:

▶ **Lemma 24** (Low-discrepancy sequence from almost normal vectors)**.** *Let $B > 0$, and $v = (v_1,\ldots,v_n)$ be some (σ,ε)-normal vector, for $\sigma = n^{-B}, \varepsilon \le n^{-0.9B}$. There exists $M \le n^{1.6B}$ such that for any $S = \{i_1,\ldots,i_s\} \subseteq [n]$, $|S| = s$ the distribution on s-dimensional sequence of length M:*

$$V_s \equiv \{(\{m\cdot v_{i_1}\},\ldots,\{m\cdot v_{i_s}\})\}_{m\in[M]}$$

satisfies $D_M(V_s) \le 4\log^s(n)\cdot n^{-0.1B}$.

Proof. Let P be the minimal prime which is at least $n^{0.3B}$, and put $M = P^5$. By Bertrand's postulate, for sufficiently large n we have that $M = P^5 \le n^{1.51B} \le n^{1.6B}$. For any $z \in [0,1)$ let z^M be the number closest to z in the grid m/M, $m \in [M]$.

Removal of non-independent component

Since v is (σ,ε)-normal then $v_i = X_i + Y_i$, where $X_i \sim (\eta_i,\sigma^2), |Y_i| \le \varepsilon\sigma$, and the X_i's are independent. Let V_S^X denote the sequence generated by taking only the X component of the seed vector v, i.e.:

$$V_S^X \equiv \{(\{m\cdot X_{i_1}\},\ldots,\{m\cdot X_{i_s}\})\}_{m\in[M]} \tag{2}$$

▶ **Fact 25.**

$$D_M(V_S) \le D_M(V_S^X) + s\cdot n^{-0.2B}$$

Proof. Consider the r.v.'s X_i, Y_i. By our assumption

$$\forall i\in[n] \quad |Y_i| \le \sigma\varepsilon = n^{-1.9B}. \tag{3}$$

Thus the difference between the residuals of v_i and X_i are small modulo 1:

$$\forall m\in[M],\ i\in[n] \quad |\lceil\{mv_i\} - \{mX_i\}\rfloor| \le m\cdot n^{-1.9B} \le Mn^{-1.9B} \le n^{-0.3B} \tag{4}$$

By Lemma 18, we can conclude that the discrepancy of our target sequence V_S follows tightly the discrepancy of V_S^X:

$$D_M(V_S) \le D_M(V_S^X) + s\cdot n^{-0.3B} \tag{5}$$

◀

Reducing Gaussian measure to uniform measure

Consider the vector derived by truncating each coordinate of the vector $(X_{i_1},\ldots,X_{i_s})$ to the nearest point on the M-grid:

$$X^M = (X_{i_1}^M,\ldots,X_{i_s}^M) = (\lfloor MX_{i_1}\rceil/M,\ldots,\lfloor MX_{i_s}\rceil/M).$$

Consider the discrepancy of the distribution on s-dimensional sequences formed by taking integer multiples of X^M. We claim:

▶ **Fact 26.**

$$\mathsf{P}_v\left(D_M(V_S^{X,M}) \le \log^s(n)\cdot n^{-0.1B}\right) \ge 1-3n^{-0.1B},$$

Proof. In Fact 22 choose as parameter $m = n^{0.2B+2}$. We get that w.p. at least $1-2n^2/m = 1-2n^{-0.2B}$ each X_i samples a convex mixture of variables $\{w_j\}_{j\in[m]}$ where

$$w_j \sim U(I_j), |I_j| = \sigma/m = n^{-1.2B-2} \tag{6}$$

Hence, w.p. at least $1-2n^{-0.2B}$ for all $i\in[n]$, the variable $M\cdot\{X_i^M\}$ is a convex mixture of uniform random variables on intervals $M\cdot I_j \subseteq [M]$, where

$$|M\cdot I_j| \ge \frac{\sigma M}{m} \ge n^{1.5B}\cdot n^{-1.2B-2} \ge M^{0.2}. \tag{7}$$

We apply Lemma 21 to the sequence of residuals of integer multiples, with the seed X^M:

$$V_S^{X,M} \equiv \left(\left\{mX_1^M\right\},\ldots,\left\{mX_s^M\right\}\right)_{m\in[M]}. \tag{8}$$

The lemma requires that each variable be distributed as: $MX_i^M \sim U[\mathcal{I}]$, where $\mathcal{I}$ is some interval of $[M]$, for integer $M>1$ satisfying: $|\mathcal{I}|\ge P,\quad P$ prime , $\quad P|N$. By our choice of parameters M has a prime divisor P equal to $M^{0.2}=P$. Hence, by Equation 7 we can satisfy the assumption of the lemma by choosing the parameters for Lemma 21 as follows: $N=M, M=P$. Hence, by Lemma 21, and accounting for the Gaussian-to-uniform approximation error we get:

$$\mathsf{P}_v\left(D_M(V_S^{X,M}) \le 2\log^s(n)\cdot n^{-0.1B}\right) \ge 1-n^{-0.1B}-2n^{-0.2B} \ge 1-3n^{-0.1B}. \tag{9}$$

◀

Treating the residual w.r.t. the M-grid

Define: the truncation error

$$\forall i\in[s]\quad r_i := X_i - X_i^M.$$

In Fact 25 we analyzed the error Y_i whose magnitude is negligible even w.r.t. $1/M$, and can thus be disregarded for any element of the sequence V_S. Unlike this, the residual error r_i cannot be disregarded because when multiplied by integers uniformly in $[M]$ it assumes magnitude $\Omega(1)$. Thus, it requires a different treatment.

▶ **Corollary 27.**

$$\mathsf{P}_v\left(D_M(V_S^X) \le 2\log^s(n)\cdot n^{-0.1B}\right) \ge 1-4n^{-0.1B}$$

Proof. Express the i-th element of the sequence using r_i:

$$\forall i\in[s]\quad \{X_i\cdot m\} = \left\{(X_i^M+r_i)\cdot m\right\} = \{\{mX_i^M\}+\{mr_i\}\} \tag{10}$$

The variable $V_S^{X,M}$ is the distribution on s-dimensional vectors formed by sampling the initial seed $\{X_i^M\}_{i\in[s]}$, a uniform random m and returning $(\{mX_1^M\},\ldots,\{mX_s^M\})\in[0,1)^s$. Hence, the variable $y\sim V_S^X$ can be written as

$$y = x + z(\mathrm{mod}1)$$

where $x \sim V_S^{X,M}$ and $z \sim \{(mr_1, \ldots, mr_s)\}$, where $m \sim U[M]$.

Let E denote the event in which X_i is sampled according to $w_j \sim U[\mathcal{I}_j]$ where w_j is at distance at least $1/M$ from either one of the edges of $\mathcal{I}_j$. Conditioned on E, the random variables r_i and X_i^M are independent for all $i \in [s]$. By the above, x and z are independent conditioned on E. Hence, we can invoke Fact 19 w.r.t. y. By this fact we have:

$$D_M(V_S^X|E) \leq D_M(V_S^{X,M})$$

and so by Fact 26

$$\mathsf{P}_v\left(D_M(V_S^X|E) \leq \log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 3n^{-0.1B}, \tag{11}$$

By Equation 7 the probability of E is at least:

$$\mathsf{P}_v(E) \geq 1 - |\mathcal{I}_j|/(2M) \geq 1 - M^{0.2}/(2M) \geq 1 - n^{-B}.$$

Thus: $\mathsf{P}_v\left(D_M(V_S^X) \leq \log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 3n^{-0.1B} - \mathsf{P}(E) \geq 1 - 4n^{-0.1B}$. ◀

Conclusion of proof: By Corollary 27 we have

$$\mathsf{P}_v\left(D_M(V_S^X) \leq 2\log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 4n^{-0.1B}$$

and by Fact 25 we have

$$D_M(V_S) \leq D_M(V_S^X) + s \cdot n^{-0.2B}$$

Thus by the union bound:

$$\mathsf{P}_v\left(D_M(V_S) \leq 2\log^s(n) \cdot n^{-0.1B} + s \cdot n^{-0.2B}\right) \geq 1 - 4n^{-0.1B}$$

thus:

$$\mathsf{P}_v\left(D_M(V_S) \leq 3\log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 4n^{-0.1B}$$

Hence for all but a measure $4n^{-0.1B}$ of sampled vectors v, the resulting sequence has discrepancy at most $3\log^s(n)n^{-0.1B}$. Since the discrepancy measures the worst-case additive error for any set this implies that:

$$D_M(V_S) \leq 3\log^s(n)n^{-0.1B} + 4n^{-0.1B} \leq 4\log^s(n)n^{-0.1B}$$ ◀

5 A Filtering Algorithm

In this section we provide the specification of the filtering algorithm, which is the main computational black box of our algorithm. This algorithm accepts an integer m that separates the i-th eigenvalue of a Hermitian matrix A and computes an approximation for the i-th eigenvector, with high probability:

Algorithm 2 Filter(A, m, δ)

1. Compute parameters: $p = 2n^2\lceil \ln(1/\delta)\rceil, \zeta = \delta^2/(2pm)$.
2. **Sample random unit vector:**
   Sample a standard complex Gaussian vector v, set $w_0 = v/\|v\|$.
3. **Approximate matrix exponent:**
   Compute a ζ Taylor-series approximation of $e^{2\pi iA}$, denoted by $\tilde{U}$.
4. **Raise to power:**
   Compute $\tilde{U}^m$ by repeated squaring.
5. **Generate matrix polynomial:**
   Compute $B = \left(\frac{I+\tilde{U}^m}{2}\right)^p$ by repeated squaring.
6. **Filter:**
   Compute $w = \frac{B\cdot w_0}{\|B\cdot w_0\|}$.
7. **Decide:**
   Set $z = A\cdot w$, $i_0 = \arg\max_{i\in[n]} |w_i|$ and compute $c = z_{i_0}/w_{i_0}$. If

   $$\|A\cdot w - c\cdot w\| \le 3\delta\sqrt{n}$$

   return w, and otherwise reject.

We now show that if the algorithm is provided with an integer m that separates the k-th eigenvalue of A in the sense defined in Definition 6, then the output is close to the k-th eigenvector of A.

▶ **Theorem 28.** *Let n be some integer, $\delta \le n^{-10}$ and $\alpha = 3\sqrt{\ln(1/\delta)}$. We are given an $n\times n$ Hermitian matrix A with eigenvalues $\{\lambda_i\}_{i\in[n]}$. Additionally, we are provided an integer m that separates k in A, w.r.t. $B_{in}(\alpha), B_{out}$, in the sense of Definition 5. Let $w = Filter(A, m, \delta)$. Then*

$$\mathsf{P}\left(\|w - v_k\| \le \delta\right) \ge 1 - 3n^{-3},$$

for some unit eigenvector v_k of λ_k, and sufficiently large n. The algorithm has boolean complexity $O(n^\omega \cdot \log(2p^2m^2/\delta^2))$, and runs in parallel time $O(\log^2(n))$.

Proof. Let $\{\tau_\ell\}_{\ell\in[n]}$ denote the set of eigenvalues of $\tilde{U}$. Since $\tilde{U}$ is a polynomial in A (truncated Taylor series) then $\{v_\ell\}_{\ell\in[n]}$ is also an orthonormal basis for $\tilde{U}$. Since in addition $\|\tilde{U} - e^{2\pi iA}\| \le \zeta$ then

$$\forall \ell\in[n] \quad \left|\tau_\ell - e^{2\pi i\lambda_\ell}\right| \le \zeta. \tag{12}$$

Let $w' = B\cdot w_0$ and denote $w_0 = \sum_{\ell\in[n]}\beta_\ell v_\ell$, and $w' = \sum_{\ell\in[n]}\alpha_\ell v_\ell$. Since $A, \tilde{U}$ share the same basis of eigenvectors, then by the definition of the matrix B the coefficients α_ℓ, β_ℓ are related by:

$$|\alpha_\ell|^2 = |\beta_\ell|^2 \cdot \left|\frac{1+{\tau_\ell}^m}{2}\right|^{2p}.$$

So by Equation 12

$$\frac{|\alpha_\ell|^2}{|\beta_\ell|^2} \ge \left|\frac{1+e^{2\pi im\lambda_\ell}}{2}\right|^{2p} - 2pm\zeta = |\cos(2\pi m\lambda_\ell/2)|^{2p} - 2pm\zeta$$

Since m separates k then $\{m\lambda_k\} \in B_{in}$, and for all $\ell \neq k$ we have $\{m\lambda_\ell\} \notin B_{out}$. Thus, for $\ell = k$:

$$\frac{|\alpha_k|^2}{|\beta_k|^2} \geq \left|\cos(2\pi/6n\sqrt{\ln(1/\delta)})\right|^{2p} - 2pm\zeta$$

Using Claim 32

$$\geq \left(1 - \frac{1}{n^2 \ln(1/\delta)}\right)^{2p} - 2pm\zeta \geq \frac{1}{2e^4}. \tag{13}$$

On the other hand, for all $\ell \neq k$ we have:

$$\frac{|\alpha_\ell|^2}{|\beta_\ell|^2} \leq \left|\frac{1 + e^{2\pi i m\lambda_\ell}}{2}\right|^{2p} + 2pm\zeta.$$

so since m separates k then the above is at most:

$$\leq |\cos(2\pi/2n)|^{2p} + 2pm\zeta$$

which by Claim 32 is at most:

$$\leq (1 - \pi^2/(3n^2))^{2n^2 \ln(1/\delta)} + 2pm\zeta \leq e^{-2\ln(1/\delta)} + 2pm\zeta \leq 2\delta^2. \tag{14}$$

By Fact 31 for any $\varepsilon = 1/\mathrm{poly}(n)$ there exists a constant $c > 0$ such that

$$\mathsf{P}(\forall i,j \quad |\beta_j| \leq c|\beta_i|\sqrt{\ln(1/\varepsilon)}/\varepsilon) \geq 1 - 3n\varepsilon.$$

Choose $\varepsilon = n^{-4}$. Then by Equations 13 and 14:

$$\mathsf{P}\left(\forall \ell \neq k \quad \frac{|\alpha_\ell|^2}{|\alpha_k|^2} \leq c^2(2\delta^2) \cdot (4e^8) \cdot 4\ln n \cdot n^8\right) \geq 1 - 3n^{-3}.$$

and so for $\delta \leq n^{-10}$ there exists $\eta \in \mathbb{C}$, $|\eta| = 1$ such that

$$\left\| \frac{w'}{\|w'\|} - \eta \cdot v_k \right\|^2 \leq \frac{1}{|\alpha_k|^2} \sum_{j \neq k} |\alpha_j|^2 \leq 32c^2 n^9 \ln n \delta^2 e^8 < \delta.$$

for sufficiently large n. Using Claim 30 we conclude that w.p. at least $1 - 3n^{-3}$ over choices w_0, the criterion is met and the algorithm returns a vector $w = w'/\|w'\|$ satisfying the equation above.

Arithmetic run-time: The approximation of $e^{2\pi i A}$ by $\tilde{U}$ requires, using Fact 33 a time at most

$$O(n^\omega \log(1/\zeta)) = O(n^\omega \cdot \log(2pm/\delta^2)).$$

Next, the repeated powering of $\tilde{U}$ to a power m requires time at most: $O(n^\omega \lceil \log(m) \rceil)$ and the repeated powering of B to the power p requires time at most: $O(n^\omega \lceil \log(p) \rceil)$ Hence the total complexity is : $O(n^\omega \cdot \log(pm/\delta^2))$.

Depth complexity: Each matrix product can be carried out in depth $\log(n)$. Each of steps 3 to 6 involves at most $\log(m)+\log(p)$ sequential matrix multiplications. Hence the depth complexity of the entire circuit is at most $\log(n)\cdot(\log(m)+\log(p))+O(\log(n))=O(\log^2(n))$.

We conclude the proof of the theorem by showing stability:

▶ **Claim 29.** *Under the assumption of Theorem 28 the algorithm is log-stable.*

Proof. Consider the arithmetic operations involved in computing the filtering algorithm:

1. Generating an approximation $\tilde{U}$ of $e^{2\pi iAm}$ as a truncated Taylor series.
2. Raising $\tilde{U}$ to a power $m\in[M]$.
3. Computing $((I+\tilde{U})/2)^p$.
4. Normalizing $Bw_0/\|Bw_0\|$.

Consider an arithmetic circuit C implementing the above, and the circuit $D=D(C,t)$ - the discretization of C to t bits of precision modeled as follows: after each arithmetic step, the result is rounded to the nearest value of 2^{-t}. Consider all steps except division. A is δ-separated so in particular $\|A\|\le 1$. Thus, whenever we multiply two matrices at any of the steps above both have norm at most 1. Hence, at each rounding step the error is increased by at most $\sqrt{n}2^{-t}$. Finally, considering the final division step, we observe that since m separates k, then by Equation 13 we have $\|Bw_0\|\ge 1/(2e^6)$. This implies that the total error is at most $\sqrt{n}(p+M)\cdot 2^{-t}\cdot 2e^6$. Since M,p are both polynomial in n then for any $\delta=1/\mathrm{poly}(n)$ the error is at most δ for some $t=O(\log(1/\delta))$. ◀

◀

5.1 Supporting Claims

We now state the important supporting claims. Their proofs appear in the full version of the paper.

▶ **Claim 30.** *Let A be some $n\times n$ Hermitian matrix, $\|A\|\le 1$. Suppose that $\|w-v\|\le\delta$ for some unit eigenvector v of A, and $\delta\le 1/4$. Let $z=A\cdot w$, and i_0 denote $i_0=\arg\max_{i\in[n]}|w_i|$. Let $c=z_{i_0}/w_{i_0}$. Then*

$$\|A\cdot w-c\cdot w\|\le 3\delta\sqrt{n}.$$

▶ **Fact 31** (Random unit vectors have well-balanced entries)**.** *Let $\{v_i\}_{i\in[n]}$ be some orthonormal basis of $\mathbb{C}^n$, $0<\varepsilon=1/\mathrm{poly}(n)$, and $v\in\mathbb{C}^n$ a uniformly random complex unit vector. For any $i\in[n]$ let $\alpha_i=|\langle v,v_i\rangle|$. For any $\varepsilon=1/\mathrm{poly}(n)$ there exists a number $c_1>0$ independent of n, such that*

$$\mathsf{P}\left(\forall i,j\quad |\alpha_i|/|\alpha_j|\le c_1\sqrt{\ln(1/\varepsilon)}/\varepsilon\right)\ge 1-3n\varepsilon.$$

▶ **Claim 32.** $\forall\theta\in[-0.01,0.01]\quad 1-\frac{\theta^2}{2}\le\frac{1+\cos(\theta)}{2}\le 1-\frac{\theta^2}{3}$.

▶ **Fact 33** (Efficient approximation of exponentiated matrix)**.** *Given a Hermitian $n\times n$ matrix A, $\|A\|\le 1/(2\pi)$, and error parameter $\varepsilon>0$, a Taylor approximation of $e^{2\pi iA}$, denoted by $\tilde{U}_A$ can be computed in time $O(n^\omega\log(1/\varepsilon))$ and satisfies $\left\|e^{2\pi iA}-\tilde{U}_A\right\|\le\varepsilon$.*

6 Sampling Separating Integers

In this section we show our main technical tool: which is that perturbing a δ-separated Hermitian matrix A by a Gaussian matrix of a carefully calibrated variance, results in a corresponding sequence of residuals $S(A)$ having low-discrepancy, at least for 2-dimensional sequences - i.e. pairs of variables. This, in turn, implies that we can separate each eigenvalue of A almost uniformly:

▶ **Theorem 34.** *Let A be a δ-separated $n \times n$ PSD matrix, $\mathcal{E}$ GUE, $\zeta \leq \min\{\delta^{13}, n^{-50}\}$, and $\alpha > 4$. For any $M \geq \zeta^{-1.6}$ we have:*

$$\forall k \in [n] \quad \mathsf{P}_{\mathcal{E}, m\sim U[M]}\left(\text{ } m \text{ separates } k \text{ in } A + \zeta \cdot \mathcal{E} \text{ w.r.t. } B_{in}(\alpha), B_{out} \text{ } \right) \geq 1/(5\alpha n)$$

6.1 Additive Perturbation

By our definitions above, Gaussian perturbation of a matrix with well-separated eigenvalues results in a (σ, ε)-normal vector as follows:

▶ **Fact 35** (Perturbation of well-separated matrices)**.** *Let A be an $n \times n$ ε-separated Hermitian matrix with eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$. Let $\mathcal{E}$ be GUE, and $A' = A + \varepsilon^L \cdot \mathcal{E}$, where $L \geq 2$. Then w.p. at least $1 - n \cdot 2^{-n}$ the vector of eigenvalues of A' $(\lambda'_1, \ldots, \lambda'_n)$ is $(\varepsilon^L, cn\varepsilon^{L-1})$-normal, for some constant $c > 0$.*

Proof. Invoke Corollary 14 choosing ε as ε^L and δ as ε, and take the union bound over all $i \in [n]$. ◀

6.2 Approximate Pairwise Independence

▶ **Lemma 36.** *Let $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n) \in [0,1)^n$ and M a positive integer that satisfy:*

$$\forall i \neq j \quad D_M\left(\{(m\lambda_i, m\lambda_j)\}_{m\in[M]}\right) \leq \zeta, \quad \zeta \leq n^{-4}$$

Let $\alpha > 4$. For each $k \in [n]$ w.p. at least $1/(5\alpha n)$ over choices of $m \sim U[M]$ the sampled sequence m separates k w.r.t. $B_{in}(\alpha), B_{out}$.

Proof. Fix $x_i = \{m\lambda_i\}$ and let E_i denote the following event:

$$E_i := (x_i \in B_{in}) \wedge (\forall j \neq i \;\; x_j \notin B_{out})$$

We want to show that

$$\forall i \in [n] \;\; \mathsf{P}(E_i) \geq \frac{1}{5\alpha n}.$$

Let t denote the number of x_j's in B_{out}:

$$t = |\{j \;\; j \neq i \;\; x_j \in B_{out}\}|$$

Then under this notation we have:

$$\mathsf{P}(E_i) = \mathsf{P}(t = 0 \wedge x_i \in B_{in}). \tag{15}$$

Consider the conditional expectation: $\mathbf{E}\left[t | x_i \in B_{in}\right]$ By linearity of expectation:

$$\mathbf{E}\left[t | x_i \in B_{in}\right] = \sum_{j\neq i} \mathsf{P}\left[x_j \in B_{out} | x_i \in B_{in}\right]. \tag{16}$$

Considering each summand separately:

$$\mathsf{P}(x_j \in B_{out} | x_i \in B_{in}) = \frac{\mathsf{P}(x_j \in B_{out} \wedge x_i \in B_{in})}{\mathsf{P}(x_i \in B_{in})}$$

Using the pairwise discrepancy assumption, the above is at most:

$$\frac{|B_{out}| \cdot |B_{in}| + \zeta}{|B_{in}| - \zeta} \leq |B_{out}| + 2\zeta\alpha n = \frac{1}{2n} + 2\alpha\zeta n \leq \frac{0.51}{n}$$

and so by Equation 16

$$\mathbf{E}\left[t | x_i \in B_{in}\right] = (n-1) \cdot \mathsf{P}\left[x_j \in B_{out} | x_i \in B_{in}\right] \leq 0.51.$$

The variable $t | x_i \in B_{in}$ accepts only integral values, and by Markov's inequality:

$$\mathsf{P}(t \geq 1 | x_i \in B_{in}) \leq 0.51$$

Therefore $\mathsf{P}(t = 0 | x_i \in B_{in}) \geq 0.49$. Using again the 1-dimensional discrepancy we have

$$\mathsf{P}(x_i \in B_{in}) \geq \frac{1}{\alpha n} - \zeta \geq \frac{1}{2\alpha n}.$$

Substituting the last two inequalities into Equation 15 yields: $\mathsf{P}(E_i) \geq 0.49 \cdot \frac{1}{2\alpha n} \geq \frac{1}{5\alpha n}$. ◀

6.3 Proof of Theorem 34

Proof. By assumption A is δ-separated and $\zeta \leq \min\{n^{-50}, \delta^{13}\}$. Consider the perturbed matrix $A' = A + \zeta\mathcal{E}$. Choose $L = 13$ and $\varepsilon = \zeta^{1/13}$. By Fact 35 there exists some constant $c > 0$ such that w.p. at least $1 - n2^{-n}$ the vector $\mathcal{L}(A')$ is (ζ, ε)-normal with parameters

$$\zeta \leq n^{-50}, \varepsilon \leq cn\zeta^{12/13} \leq \zeta^{0.9}$$

where the last inequality follows because $\zeta \leq n^{-50}$. We assume that this is the case and account for the negligible error at the end. Set now $B = \log_n(1/\zeta)$. Then the eigenvalues of A' are (σ, ε)-normal for $\sigma = n^{-B}$ an $\varepsilon \leq n^{-0.9B}$. Since in addition $\alpha > 4$ then by Lemma 24 there exists an integer $M \leq n^{1.6B}$ satisfying:

$$\forall S \subseteq [n],\ |S| = s \quad D_M(\{m\lambda_S\}) \leq 4\log^s(n)n^{-5} \leq n^{-4}, \tag{17}$$

for sufficiently large n. Hence, by Lemma 36 a random $m \sim U[M]$ separates the k-th eigenvalue of $A + \mathcal{E}$ w.r.t. $B_{in}(\alpha), B_{out}$ w.p at least $1/(5\alpha n)$. ◀

7 Parallel Algorithm for ASD

The algorithm Filter(A, m, δ) described in Section 5 is given an integer m that separates the i-th eigenvalue, and returns an approximation for the i-th eigenvector. In this section, we use this algorithm in a black-box fashion and design a Las-Vegas algorithm for computing the full ASD of a matrix. Essentially, it amounts to running sufficiently many copies of the filtering algorithm in parallel so that all eigenvectors are collected as "coupons" with high probability.

Algorithm 3

Input: $n \times n$ Hermitian matrix A, parameter δ.

1. **Initialize:**
   a. Recall the definition of B^* in Definition 16 and compute parameters: $\delta = \min\{\delta, n^{-10}\}$, $B = \min\{\delta, B^*(\delta/(3\sqrt{n}))\}$, $\delta' = (\min\{\delta, B\})^{13}/4$, $\alpha = \sqrt{\ln(1/\delta')}$, $M = (\max\{B^{-12}, n^{-50}\})^{1.6}$, $\mathcal{T} = 60n\alpha\log(n)$.
   b. Perturb A with GUE matrix $\mathcal{E}_1$: $A_0 := A + \mathcal{E}_1 \cdot \delta/(3n)$
2. **Collect all eigenvectors:**
   Run $\mathcal{T}$ parallel processes of the following procedure
   a. Perturb A_0: $A_1 = A_0 + \delta' \cdot \mathcal{E}_2$, for GUE matrix $\mathcal{E}_2$.
   b. Sample $m \sim U[M]$
   c. Run Filter (A_1, m, δ') and store output w.
3. **Generate database:**
   a. For vector $w = w_k$ sampled at process $i \in [\mathcal{T}]$, compute $z = A \cdot w$, $i_0 = \arg\max_{i \in [n]} |w_i|$ and $\tilde{\lambda}_k = z_{i_0}/w_{i_0}$.
   b. Sort the values $\tilde{\lambda}_i$: assume $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_{\mathcal{T}}$. Initialize: $\gamma = \tilde{\lambda}_1$, $\mathcal{D} = \emptyset$. Iterate over all $i = 1, \ldots, \mathcal{T}$. At each step i: if $|\gamma - \tilde{\lambda}_i| \geq B/4$ then add $\mathcal{D} \to \mathcal{D} \cup \{w_i\}$, and set $\gamma = \tilde{\lambda}_i$.

Overview:

The first step of the algorithm adds a "coarse" perturbation to A to make sure that it has n unique eigenvalues that are well-separated. The second step is essentially a parallel execution of the Filter() procedure where each call to this sub-routine uses independent random bits to add a "fine" perturbation to A. This implies that each process samples independently and uniformly an approximation of the k-th eigenvector of A, for each $k \in [n]$. The final step merely builds up a database of approximate eigenvectors so that all eigenvectors are represented exactly once.

We now state our main theorem the proof of which appears in the full version of the paper:

▶ **Theorem 37.** *For any $n \times n$ Hermitian matrix $0 \preceq A \preceq 0.9I$, and $\delta = 1/\mathrm{poly}(n)$ there exists an $\mathrm{RNC}^{(2)}$ algorithm computing $\mathrm{ASD}(A, \delta)$, in boolean complexity $\tilde{O}(n^{\omega+1})$. The algorithm is log-stable.*

Acknowledgements. The authors thank Naomi Kirshner, Robin Kothari, Yosi Atia, and anonymous reviewers for their helpful comments and suggestions.

References

1 Selim G. Akl. *Parallel Sorting Algorithms*. Academic Press, Inc., Orlando, FL, USA, 1990.
2 F. L. Bauer and C. T. Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2(1):137–141, Dec 1960. doi:10.1007/BF01386217.
3 Dario Bini and Victor Y. Pan. Practical improvement of the divide-and-conquer eigenvalue algorithms. *Computing*, 48(1):109–123, 1992. doi:10.1007/BF02241709.
4 James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007. doi:10.1007/s00211-007-0114-x.

5 Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

6 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009. doi:10.1103/PhysRevLett.103.150502.

7 Dexter C. Kozen. *The Design and Analysis of Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.

8 Mauro Leoncini, Giovanni Manzini, and Luciano Margara. Parallel complexity of numerically accurate linear system solvers. *SIAM J. Comput.*, 28(6):2030–2058, 1999. doi:10.1137/S0097539797327118.

9 Hoi Nguyen, Terence Tao, and Van Vu. Random matrices: tail bounds for gaps between eigenvalues. *Probability Theory and Related Fields*, pages 1–40, 2016. doi:10.1007/s00440-016-0693-5.

10 H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992. doi:10.1137/1.9781611970081.

11 John H. Reif. Efficient parallel factorization and solution of structured and unstructured linear systems. *Journal of Computer and System Sciences*, 71(1):86–143, 2005.

12 Gilbert W. Stewart and Jiguang Sun. *Matrix perturbation theory*. Computer science and scientific computing. Academic Press, Boston, 1990.

13 G.W. Stewart. Pertubation bounds for the definite generalized eigenvalue problem. *Linear Algebra and its Applications*, 23:69–85, 1979.

14 Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 881–890. ACM, 2013. doi:10.1145/2488608.2488720.

15 Lloyd N. Trefethen and David Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

16 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 887–898. ACM, 2012. doi:10.1145/2213977.2214056.