2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)

Pseudospectral Shattering, the Sign Function, and Diagonalization in Nearly Matrix Multiplication Time

Jess Banks
Mathematics
UC Berkeley
jess.m.banks@berkeley.edu

Jorge Garza-Vargas
Mathematics
UC Berkeley
jgarzagargas@berkeley.edu

Archit Kulkarni
Mathematics
UC Berkeley
akulkarni@berkeley.edu

Nikhil Srivastava
Mathematics
UC Berkeley
nikhil@math.berkeley.edu

***Abstract*—We exhibit a randomized algorithm which given a square matrix $A \in \mathbb{C}^{n\times n}$ with $\|A\| \leq 1$ and $\delta > 0$, computes with high probability an invertible V and diagonal D such that**

$$\|A - VDV^{-1}\| \leq \delta$$

in $O(T_{\mathsf{MM}}(n)\log^2(n/\delta))$ arithmetic operations on a floating point machine with $O(\log^4(n/\delta)\log n)$ bits of precision. The computed similarity V additionally satisfies $\|V\|\|V^{-1}\| \leq O(n^{2.5}/\delta)$. Here $T_{\mathsf{MM}}(n)$ is the number of arithmetic operations required to multiply two $n \times n$ complex matrices numerically stably, known to satisfy $T_{\mathsf{MM}}(n) = O(n^{\omega+\eta})$ for every $\eta > 0$ where ω is the exponent of matrix multiplication [1]. The algorithm is a variant of the spectral bisection algorithm in numerical linear algebra [2] with a crucial Gaussian perturbation preprocessing step. Our running time is optimal up to polylogarithmic factors, in the sense that verifying that a given similarity diagonalizes a matrix requires at least matrix multiplication time. It significantly improves the previously best known provable running times of $O(n^{10}/\delta^2)$ arithmetic operations for diagonalization of general matrices [3], and (with regards to the dependence on n) $O(n^3)$ arithmetic operations for Hermitian matrices [4], and is the first algorithm to achieve nearly matrix multiplication time for diagonalization in any model of computation (real arithmetic, rational arithmetic, or finite arithmetic).

The proof rests on two new ingredients. (1) We show that adding a small complex Gaussian perturbation to *any* matrix splits its pseudospectrum into n small well-separated components. In particular, this implies that the eigenvalues of the perturbed matrix have a large minimum gap, a property of independent interest in random matrix theory. (2) We give a rigorous analysis of Roberts' [5] Newton iteration method for computing the sign function of a matrix in finite arithmetic, itself an open problem in numerical analysis since at least 1986 [6]. This is achieved by controlling the evolution of the pseudospectra of the iterates using a carefully chosen sequence of shrinking contour integrals in the complex plane.

***Keywords*-Numerical Analysis, Random Matrix Theory.**

I. Introduction

We study the algorithmic problem of approximately finding all of the eigenvalues and eigenvectors of a given arbitrary $n \times n$ complex matrix. While this problem is quite well-understood in the special case of Hermitian matrices (see, e.g., [4]), the general non-Hermitian case has remained mysterious from a theoretical standpoint even after several decades of research. In particular, the currently best known *provable* algorithms for this problem run in time $O(n^{10}/\delta^2)$ [3] or $O(n^c \log(1/\delta))$ [7] with $c \geq 12$ where $\delta > 0$ is an error parameter, depending on the model of computation and notion of approximation considered.[1] To be sure, the non-Hermitian case is well-motivated: coupled systems of differential equations, linear dynamical systems in control theory, transfer operators in mathematical physics, and the nonbacktracking matrix in spectral graph theory are but a few situations where finding the eigenvalues *and eigenvectors* of a non-Hermitian matrix is important.

The key difficulties in dealing with non-normal matrices are the interrelated phenomena of *non-orthogonal eigenvectors* and *spectral instability*, the latter referring to extreme sensitivity of the eigenvalues and invariant subspaces to perturbations of the matrix. Non-orthogonality slows down convergence of standard algorithms such as the power method, and spectral instability can force the use of very high precision arithmetic, also leading to slower algorithms. Both phenomena together make it difficult to reduce the eigenproblem to a subproblem by "removing" an eigenvector or invariant subspace, since this can only be done approximately and one must control the spectral stability of the subproblem.

In this paper, we overcome these difficulties by identifying and leveraging a phenomenon we refer to as *pseudospectral shattering*: adding a small complex Gaussian perturbation to any matrix yields a matrix with well-conditioned eigenvectors and a large minimum gap between the eigenvalues, implying spectral stability. This result builds on the recent solution of Davies' conjecture [8], and is of independent interest in random matrix theory, where minimum eigenvalue gap bounds in the non-Hermitian case were previously only known for i.i.d. models [9], [10].

We complement the above by proving that a variant of the well-known spectral bisection algorithm in numerical linear algebra [2] is both fast and numerically stable (i.e., can be implemented using a polylogarithmic number of bits of

[1] A detailed discussion of these and other related results appears in Section I-C.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00056

precision) when run on a pseudospectrally shattered matrix. The key step in the bisection algorithm is computing the *sign function* of a matrix, a problem of independent interest in many areas such including control theory and approximation theory [11]. Our main algorithmic contribution is a rigorous analysis of the well-known Newton iteration method [5] for computing the sign function *in finite arithmetic*, showing that it converges quickly and numerically stably on matrices for which the sign function is well-conditioned, in particular on pseudospectrally shattered ones.

The end result is an algorithm which reduces the general diagonalization problem to a polylogarithmic (in the desired accuracy and dimension n) number of invocations of standard numerical linear algebra routines (multiplication, inversion, and QR factorization), each of which is reducible to matrix multiplication [12], yielding a nearly matrix multiplication runtime for the whole algorithm. This improves on the previously best known running time of $O(n^3+n^2\log(1/\delta))$ arithmetic operations even in the Hermitian case [4].

We now proceed to give precise mathematical formulations of the eigenproblem and computational model, followed by statements of our results and a detailed discussion of related work.

A. *Problem Statement*

An *eigenpair* of a matrix $A \in \mathbb{C}^{n\times n}$ is a tuple $(\lambda, v) \in \mathbb{C}\times\mathbb{C}^n$ such that

$$Av = \lambda v,$$

and v is normalized to be a unit vector. The *eigenproblem* is the problem of finding a maximal set of linearly independent eigenpairs (λ_i, v_i) of a given matrix A; note that an eigenvalue may appear more than once if it has geometric multiplicity greater than one. In the case when A is diagonalizable, the solution consists of exactly n eigenpairs, and if A has distinct eigenvalues then the solution is unique, up to the phases of the v_i.

1) Accuracy and Conditioning: Due to the Abel-Ruffini theorem, it is impossible to have a finite-time algorithm which solves the eigenproblem exactly using arithmetic operations and radicals. Thus, all we can hope for is *approximate* eigenvalues and eigenvectors, up to a desired accuracy $\delta > 0$. There are two standard notions of approximation. We assume $\|A\| \leq 1$ for normalization, where throughout this work, $\|\cdot\|$ denotes the spectral norm (the $\ell^2 \to \ell^2$ operator norm).

Forward Approximation. Compute pairs (λ_i', v_i') such that

$$|\lambda_i - \lambda_i'| \leq \delta \quad \text{and} \quad \|v_i - v_i'\| \leq \delta$$

for the true eigenpairs (λ_i, v_i), i.e., find a solution close to the exact solution. This makes sense in contexts where the exact solution is meaningful; e.g. the matrix is of theoretical/mathematical origin, and unstable (in the entries) quantities such as eigenvalue multiplicity can have a significant meaning.

Backward Approximation. Compute (λ_i', v_i') which are the exact eigenpairs of a matrix A' satisfying

$$\|A' - A\| \leq \delta,$$

i.e., find the exact solution to a nearby problem. This is the appropriate and standard notion in scientific computing, where the matrix is of physical or empirical origin and is not assumed to be known exactly (and even if it were, roundoff error would destroy this exactness). Note that since diagonalizable matrices are dense in $\mathbb{C}^{n\times n}$, one can hope to always find a complete set of eigenpairs for some nearby $A' = VDV^{-1}$, yielding an *approximate diagonalization* of A:

$$\|A - VDV^{-1}\| \leq \delta. \tag{1}$$

Note that the eigenproblem in either of the above formulations is *not* easily reducible to the problem of computing eigenvalues, since they can only be computed approximately and it is not clear how to obtain approximate eigenvectors from approximate eigenvalues. We now introduce a condition number for the eigenproblem, which measures the sensitivity of the eigenpairs of a matrix to perturbations and allows us to relate its forward and backward approximate solutions.

Condition Numbers. For diagonalizable A, the *eigenvector condition number* of A, denoted $\kappa_V(A)$, is defined as:

$$\kappa_V(A) := \inf_V \|V\|\|V^{-1}\|, \tag{2}$$

where the infimum is over all invertible V such that $A = VDV^{-1}$ for some diagonal D, and its *minimum eigenvalue gap* is defined as:

$$\mathrm{gap}(A) := \min_{i\neq j} |\lambda_i(A) - \lambda_j(A)|,$$

where λ_i are the eigenvalues of A (with multiplicity). We define the *condition number of the eigenproblem* to be[2]:

$$\kappa_{\mathrm{eig}}(A) := \frac{\kappa_V(A)}{\mathrm{gap}(A)} \in [0, \infty]. \tag{3}$$

It follows from the following proposition (whose proof appears in [14]) that a δ-backward approximate solution of the eigenproblem is a $6n\kappa_{\mathrm{eig}}(A)\delta$-forward approximate solution[3]

[2] This quantity is inspired by but not identical to the "reciprocal of the distance to ill-posedness" for the eigenproblem considered by Demmel [13], to which it is polynomially related.

[3] In fact, it can be shown that $\kappa_{\mathrm{eig}}(A)$ is related by a $\mathrm{poly}(n)$ factor to the smallest constant for which (4) holds for all sufficiently small $\delta > 0$.

Proposition I.1. *If $\|A\|, \|A'\| \leq 1$, $\|A - A'\| \leq \delta$, and $\{(v_i, \lambda_i)\}_{i \leq n}$, $\{(v_i', \lambda_i')\}_{i \leq n}$ are eigenpairs of A, A' with distinct eigenvalues, and $\delta < \frac{\mathrm{gap}(A)}{8\kappa_V(A)}$, then*

$$\|v_i' - v_i\| \leq 6n\kappa_{\mathrm{eig}}(A)\delta \text{ and } \tag{4}$$

$$\|\lambda_i' - \lambda_i\| \leq \kappa_V(A)\delta \leq 2\kappa_{\mathrm{eig}}(A)\delta \quad \forall i = 1, \ldots, n, \tag{5}$$

after possibly multiplying the v_i by phases.

Note that $\kappa_{\mathrm{eig}} = \infty$ if and only if A has a double eigenvalue; in this case, a relation like (4) is not possible since different infinitesimal changes to A can produce macroscopically different eigenpairs.

In this paper we will present a backward approximation approximation for the eigenproblem with running time scaling polynomially in $\log(1/\delta)$, which by (4) yields a forward approximation algorithm with running time scaling polynomially in $\log(1/\kappa_{\mathrm{eig}}\delta)$.

Remark I.2 (Multiple Eigenvalues)**.** A backward approximation algorithm for the eigenproblem can be used to accurately find bases for the eigenspaces of matrices with multiple eigenvalues, but quantifying the forward error requires introducing condition numbers for invariant subspaces rather than eigenpairs. A standard treatment of this can be found in any numerical linear algebra textbook, e.g. [15], and we do not discuss it further in this paper for simplicity of exposition.

2) Models of Computation: These questions may be studied in various computational models: exact *real arithmetic* (i.e., infinite precision), *variable precision rational arithmetic* (rationals are stored exactly as numerators and denominators), and *finite precision arithmetic* (real numbers are rounded to a fixed number of bits which may depend on the input size and accuracy). Only the last two models yield actual Boolean complexity bounds, but introduce a second source of error stemming from the fact that computers cannot exactly represent real numbers.

We study the third model in this paper, axiomatized as follows.

Finite Precision Arithmetic. We use the standard axioms from [16]. Numbers are stored and manipulated approximately up to some machine precision $\mathbf{u} := \mathbf{u}(\delta, n) > 0$, which for us will depend on the instance size n and desired accuracy δ. This means every number $x \in \mathbb{C}$ is stored as $\mathsf{fl}(x) = (1 + \Delta)x$ for some adversarially chosen $\Delta \in \mathbb{C}$ satisfying $|\Delta| \leq \mathbf{u}$, and each arithmetic operation $\circ \in \{+, -, \times, \div\}$ is guaranteed to yield an output satisfying

$$\mathsf{fl}(x \circ y) = (x \circ y)(1 + \Delta) \quad |\Delta| \leq \mathbf{u}.$$

It is also standard and convenient to assume that we can evaluate $\sqrt{x}$ for any $x \in \mathbb{R}$, where again $\mathsf{fl}(\sqrt{x}) = \sqrt{x}(1 + \Delta)$ for $|\Delta| \leq \mathbf{u}$.

Thus, the outcomes of all operations are adversarially noisy due to roundoff. The bit lengths of numbers stored in this form remain fixed at $\lg(1/\mathbf{u})$, where $\lg$ denotes the logarithm base 2. The *bit complexity* of an algorithm is therefore the number of arithmetic operations times $O^*(\log(1/\mathbf{u}))$, the running time of standard floating point arithmetic, where the $*$ suppresses $\log\log(1/\mathbf{u})$ factors. We will state all running times in terms of arithmetic operations accompanied by the required number of bits of precision, which thereby immediately imply bit complexity bounds.

Remark I.3 (Overflow, Underflow, and Additive Error)**.** Using p bits for the exponent in the floating-point representation allows one to represent numbers with magnitude in the range $[2^{-2^p}, 2^{2^p}]$. It can be easily checked that all of the nonzero numbers, norms, and condition numbers appearing during the execution of our algorithms lie in the range $[2^{-\lg^c(n/\delta)}, 2^{\lg^c(n/\delta)}]$ for some small c, so overflow and underflow do not occur. In fact, we could have analyzed our algorithm in a computational model where every number is simply rounded to the nearest rational with denominator $2^{\lg^c(n/\delta)}$—corresponding to *additive* arithmetic errors. We have chosen to use the multiplicative error floating point model since it is the standard in numerical analysis, but our algorithms do not exploit any subtleties arising from the difference between the two models.

The advantages of the floating point model are that it is realistic and potentially yields very fast algorithms by using a small number of bits of precision (polylogarithmic in n and $1/\delta$), in contrast to rational arithmetic, where even a simple operation such as inverting an $n \times n$ integer matrix requires n extra bits of precision (see, e.g., Chapter 1 of [17]). An iterative algorithm that can be implemented in finite precision (typically, polylogarithmic in the input size and desired accuracy) is called *numerically stable*, and corresponds to a dynamical system whose trajectory to the approximate solution is robust to adversarial noise (see, e.g. [18]).

The disadvantage of the model is that it is only possible to compute forward approximations of quantities which are *well-conditioned* in the input — in particular, discontinuous quantities such as eigenvalue multiplicity cannot be computed in the floating point model, since it is not even assumed that the input is stored exactly.

B. Results and Techniques

In addition to κ_{eig}, we will need some more refined quantities measure the stability of the eigenvalues and eigenvectors of a matrix to perturbations, and to state our results regarding it. The most important of these is the ϵ-pseudospectrum, defined for any $\epsilon > 0$ and $M \in \mathbb{C}^{n \times n}$

as:

$$\Lambda_\epsilon(M) := \{\lambda \in \mathbb{C} : \lambda \in \Lambda(M+E) \text{ for some } \|E\| < \epsilon\} \tag{6}$$

$$= \left\{\lambda \in \mathbb{C} : \left\|(\lambda - M)^{-1}\right\| > 1/\epsilon\right\} \tag{7}$$

where $\Lambda(\cdot)$ denotes the spectrum of a matrix. The equivalence of (6) and (7) is simple and can be found in the excellent book [19].

Eigenvalue Gaps, κ_V, and Pseudospectral Shattering. The key probabilistic result of the paper is that a random *complex* Gaussian perturbation of any matrix yields a nearby matrix with large minimum eigenvalue gap and small κ_V.

Theorem I.4 (Smoothed Analysis of gap and κ_V)**.** *Suppose $A \in \mathbb{C}^{n\times n}$ with $\|A\| \le 1$, and $\gamma \in (0, 1/2)$. Let G_n be an $n \times n$ matrix with i.i.d. complex Gaussian $N(0, 1_{\mathbb{C}}/n)$ entries, and let $X := A + \gamma G_n$. Then*

$$\kappa_V(X) \le \frac{n^2}{\gamma}, \quad \operatorname{gap}(X) \ge \frac{\gamma^4}{n^5}, \quad \textit{and} \quad \|G_n\| \le 4,$$

with probability at least $1 - 12/n^2$.

The proof of Theorem I.4 appears in [14, §3]. The key idea is to first control $\kappa_V(X)$ using [8], and then observe that for a matrix with small κ_V, two eigenvalues of X near a complex number z imply a small *second-least* singular value of $z - X$, which we are able to control.

In Section [14, §3] we develop the notion of *pseudospectral shattering*, which is implied by Theorem I.4 and says roughly that the pseudospectrum consists of n components that lie in separate squares of an appropriately coarse grid in the complex plane. This is useful in the analysis of the spectral bisection algorithm in Section [14, §5].

Matrix Sign Function. The sign function of a number $z \in \mathbb{C}$ with $\operatorname{Re}(z) \neq 0$ is defined as $+1$ if $\operatorname{Re}(z) > 0$ and -1 if $\operatorname{Re}(z) < 0$. The *matrix sign function* of a matrix A with Jordan normal form

$$A = V \begin{bmatrix} N & \\ & P \end{bmatrix} V^{-1},$$

where N (resp. P) has eigenvalues with strictly negative (resp. positive) real part, is defined as

$$\operatorname{sgn}(A) = V \begin{bmatrix} -I_N & \\ & I_P \end{bmatrix} V^{-1},$$

where I_P denotes the identity of the same size as P. The sign function is undefined for matrices with eigenvalues on the imaginary axis. Quantifying this discontinuity, Bai and Demmel [20] defined the following condition number for the sign function:

$$\kappa_{\text{sign}}(M) := \inf\left\{1/\epsilon^2 : \Lambda_\epsilon(M) \cap \{\operatorname{Re}(z) = 0\} = \varnothing\right\}, \tag{8}$$

and gave perturbation bounds for $\operatorname{sgn}(M)$ depending on κ_{sign}.

Roberts [5] showed that the simple iteration

$$A_{k+1} = \frac{A_k + A_k^{-1}}{2} \tag{9}$$

converges globally and quadratically to $\operatorname{sgn}(A)$ in exact arithmetic, but his proof relies on the fact that all iterates of the algorithm are simultaneously diagonalizable, a property which is destroyed in finite arithmetic since inversions can only be done approximately.[4] In Section [14, §4] we show that this iteration is indeed convergent when implemented in finite arithmetic for matrices with small κ_{sign}, given a numerically stable matrix inversion algorithm. This leads to the following result:

Theorem I.5 (Sign Function Algorithm)**.** *There is a deterministic algorithm* SGN *which on input an $n \times n$ matrix A with $\|A\| \le 1$, a number K with $K \ge \kappa_{\text{sign}}(A)$, and a desired accuracy $\beta \in (0, 1/12)$, outputs an approximation* $\mathsf{SGN}(A)$ *with*

$$\|\mathsf{SGN}(A) - \operatorname{sgn}(A)\| \le \beta,$$

in

$$O((\log K + \log\log(1/\beta))T_{\mathsf{INV}}(n)) \tag{10}$$

arithmetic operations on a floating point machine with

$$\lg(1/\mathbf{u}) = O(\log n \log^3 K(\log K + \log(1/\beta)))$$

bits of precision, where $T_{\mathsf{INV}}(n)$ denotes the number of arithmetic operations used by a numerically stable matrix inversion algorithm (satisfying Definition II.7).

The main new idea in the proof of Theorem I.5 is to control the evolution of the pseudospectra $\Lambda_{\epsilon_k}(A_k)$ of the iterates with appropriately decreasing (in k) parameters ϵ_k, using a sequence of carefully chosen shrinking contour integrals in the complex plane. The pseudospectrum provides a richer induction hypothesis than scalar quantities such as condition numbers, and allows one to control all quantities of interest using the holomorphic functional calculus. This technique is introduced in Sections [14, §4.1] and [14, §4.2], and carried out in finite arithmetic in Section [14, §4.3], yielding Theorem I.5.

Diagonalization by Spectral Bisection. Given an algorithm for computing the sign function, there is a natural and well-known approach to the eigenproblem pioneered in [2]. The idea is that the matrices $(I \pm \operatorname{sgn}(A))/2$ are spectral projectors onto the invariant subspaces corresponding to the eigenvalues of A in the left and right open half planes, so if some shifted matrix $z + A$ or $z + iA$ has roughly half its eigenvalues in each half plane, the problem can be reduced to smaller subproblems appropriate for recursion.

[4] Doing the inversions exactly in rational arithmetic could require numbers of bit length n^k for k iterations, which will typically not even be polynomial.

The two difficulties in carrying out the above approach are: (a) efficiently computing the sign function (b) finding a balanced splitting along an axis that is well-separated from the spectrum. These are nontrivial even in exact arithmetic, since the iteration (9) converges slowly if (b) is not satisfied, even without roundoff error. We use Theorem I.4 to ensure that a good splitting always exists after a small Gaussian perturbation of order δ, and Theorem I.5 to compute splittings efficiently in finite precision. Combining this with well-understood techniques such as rank-revealing QR factorization, we obtain the following theorem, whose proof appears in Section [14, §5].

Theorem I.6 (Backward Approximation Algorithm)**.** *There is a randomized algorithm* EIG *which on input any matrix* $A \in \mathbb{C}^{n\times n}$ *with* $\|A\| \leq 1$ *and a desired accuracy parameter* $\delta > 0$ *outputs a diagonal* D *and invertible* V *such that*

$$\|A - VDV^{-1}\| \leq \delta \quad \text{and} \quad \kappa(V) \leq 32n^{2.5}/\delta$$

in

$$O\left(T_{\mathsf{MM}}(n)\log^2\frac{n}{\delta}\right)$$

arithmetic operations on a floating point machine with

$$O(\log^4(n/\delta)\log n)$$

bits of precision, with probability at least $1 - 1/n - 12/n^2$*. Here* $T_{\mathsf{MM}}(n)$ *refers to the running time of a numerically stable matrix multiplication algorithm (detailed in Section II-E).*

Since there is a correspondence in terms of the condition number between backward and forward approximations, and as it is customary in numerical analysis, our discussion revolves around backward approximation guarantees. For convenience of the reader we write down below the explicit guarantees that one gets by using (4) and invoking EIG with accuracy $\frac{\delta}{6n\kappa_{\mathrm{eig}}}$.

Corollary I.7 (Forward Approximation Algorithm)**.** *There is a randomized algorithm which on input any matrix* $A \in \mathbb{C}^{n\times n}$ *with* $\|A\| \leq 1$*, a desired accuracy parameter* $\delta > 0$*, and an estimate* $K \geq \kappa_{\mathrm{eig}}(A)$ *outputs a* $\delta-$*forward approximate solution to the eigenproblem for* A *in*

$$O\left(T_{\mathsf{MM}}(n)\log^2\frac{nK}{\delta}\right)$$

arithmetic operations on a floating point machine with

$$O(\log^4(nK/\delta)\log n)$$

bits of precision, with probability at least $1 - 1/n - 12/n^2$*. Here* $T_{\mathsf{MM}}(n)$ *refers to the running time of a numerically stable matrix multiplication algorithm (detailed in Section II-E).*

Remark I.8 (Accuracy vs. Precision)**.** The gold standard of "backward stability" in numerical analysis postulates that

$$\log(1/\mathbf{u}) = \log(1/\delta) + \log(n),$$

i.e., the number of bits of precision is linear in the number of bits of accuracy. The relaxed notion of "logarithmic stability" introduced in [1] requires

$$\log(1/\mathbf{u}) = \log(1/\delta) + O(\log^c(n)\log(\kappa))$$

for some constant c, where κ is an appropriate condition number. In comparison, Theorem I.6 obtains the weaker relationship

$$\log(1/\mathbf{u}) = O(\log^4(1/\delta)\log(n) + \log^5(n)),$$

which is still polylogarithmic in n in the regime $\delta = 1/\mathrm{poly}(n)$.

C. Related Work

Minimum Eigenvalue Gap. The minimum eigenvalue gap of random matrices has been studied in the case of Hermitian and unitary matrices, beginning with the work of Vinson [21], who proved an $\Omega(n^{-4/3})$ lower bound on this gap in the case of the Gaussian Unitary Ensemble (GUE) and the Circular Unitary Ensemble (CUE). Bourgade and Ben Arous [22] derived exact limiting formulas for the distributions of all the gaps for the same ensembles. Nguyen, Tao, and Vu [23] obtained non-asymptotic inverse polynomial bounds for a large class of non-integrable Hermitian models with i.i.d. entries (including Bernoulli matrices).

In a different direction, Aizenman et al. proved an inverse-polynomial bound [24] in the case of an arbitrary Hermitian matrix plus a GUE matrix or a Gaussian Orthogonal Ensemble (GOE) matrix, which may be viewed as a smoothed analysis of the minimum gap. Theorem **??** may be viewed as a non-Hermitian analogue of the last result.

In the non-Hermitian case, Ge [10] obtained an inverse polynomial bound for i.i.d. matrices with real entries satisfying some mild moment conditions, and [9][5] proved an inverse polynomial lower bound for the complex Ginibre ensemble. Theorem **??** may be seen as a generalization of these results to non-centered complex Gaussian matrices.

Smoothed Analysis and Free Probability. The study of numerical algorithms on Gaussian random matrices (i.e., the case $A = 0$ of smoothed analysis) dates back to [25], [26], [27], [28]. The powerful idea of improving the conditioning of a numerical computation by adding a small amount of Gaussian noise was introduced by Spielman and Teng in [29], in the context of the simplex algorithm. Sankar, Spielman, and Teng [30] showed that adding real Gaussian

[5] At the time of writing, the work [9] is still an unpublished arXiv preprint.

noise to any matrix yields a matrix with polynomially-bounded condition number; [8] can be seen as an extension of this result to the condition number of the eigenvector matrix, where the proof crucially requires that the Gaussian perturbation is complex rather than real. The main difference between our results and most of the results on smoothed analysis (including [3]) is that our running time depends logarithmically rather than polynomially on the size of the perturbation.

The broad idea of regularizing the spectral instability of a nonnormal matrix by adding a random matrix can be traced back to the work of Śniady [31] and Haagerup and Larsen [32] in the context of Free Probability theory.

Matrix Sign Function. The matrix sign function was introduced by Zolotarev in 1877. It became a popular topic in numerical analysis following the work of Beavers and Denman [33], [2], [34] and Roberts [5], who used it first to solve the algebraic Ricatti and Lyapunov equations and then as an approach to the eigenproblem; see [11] for a broad survey of its early history. The numerical stability of Roberts' Newton iteration was investigated by Byers [6], who identified some cases where it is and isn't stable. Malyshev [35], Byers, He, and Mehrmann [36], Bai, Demmel, and Gu [37], and Bai and Demmel [20] studied the condition number of the matrix sign function, and showed that if the Newton iteration converges then it can be used to obtain a high-quality invariant subspace[6], but did not prove convergence in finite arithmetic and left this as an open question.[7] The key issue in analyzing the convergence of the iteration is to bound the condition numbers of the intermediate matrices that appear, as N. Higham remarks in his 2008 textbook:

> Of course, to obtain a complete picture, we also need to understand the effect of rounding errors on the iteration prior to convergence. This effect is surprisingly difficult to analyze. . . . Since errors will in general occur on each iteration, the overall error will be a complicated function of $\kappa_{sign}(X_k)$ and E_k for all k. . . . We are not aware of any published rounding error analysis for the computation of *sign*(A) via the Newton iteration. –[38, Section 5.7]

This is precisely the problem solved by Theorem I.5, which is as far as we know the first provable algorithm for computing the sign function of an arbitrary matrix which does not require computing the Jordan form.

In the special case of Hermitian matrices, Higham [39] established efficient reductions between the sign function and the polar decomposition. Byers and Xu [40] proved backward stability of a certain scaled version of the Newton iteration for Hermitian matrices, in the context of computing the polar decomposition. Higham and Nakatsukasa [41] (see also the improvement [42]) proved backward stability of a different iterative scheme for computing the polar decomposition, and used it to give backward stable spectral bisection algorithms for the Hermitian eigenproblem with $O(n^3)$-type complexity.

Non-Hermitian Eigenproblem. *Floating Point Arithmetic.* The eigenproblem has been thoroughly studied in the numerical analysis community, in the floating point model of computation. While there are provably fast and accurate algorithms in the Hermitian case (see the next subsection) and a large body of work for various structured matrices (see, e.g., [43]), the general case is not nearly as well-understood. As recently as 1997, J. Demmel remarked in his well-known textbook [15]: ". . . the problem of devising an algorithm [for the non-Hermitian eigenproblem] that is numerically stable and globally (and quickly!) convergent remains open."

Demmel's question remained entirely open until 2015, when it was answered in the following sense by Armentano, Beltrán, Bürgisser, Cucker, and Shub in the remarkable paper [3]. They exhibited an algorithm (see their Theorem 2.28) which given any $A \in \mathbb{C}^{n\times n}$ with $\|A\| \leq 1$ and $\sigma > 0$ produces in $O(n^9/\sigma^2)$ expected arithmetic operations the diagonalization of the nearby random perturbation $A + \sigma G$ where G is a matrix with standard complex Gaussian entries. By setting σ sufficiently small, this may be viewed as a backward approximation algorithm for diagonalization, in that it solves a nearby problem essentially exactly[8] – in particular, by setting $\sigma = \delta/\sqrt{n}$ and noting that $\|G\| = O(\sqrt{n})$ with very high probability, their result implies a running time of $O(n^{10}/\delta^2)$ in our setting. Their algorithm is based on homotopy continuation methods, which they argue informally are numerically stable and can be implemented in finite precision arithmetic. Our algorithm is similar on a high level in that it adds a Gaussian perturbation to the input and then obtains a high accuracy forward approximate solution to the perturbed problem. The difference is that their overall running time depends polynomially rather than logarithmically on the accuracy δ desired with respect to the original unperturbed problem.

Other Models of Computation. If we relax the requirements further and ask for any provable algorithm in any model of Boolean computation, there is only one more positive result with a polynomial bound on the number of bit operations: Jin Yi Cai showed in 1994 [7] that given a rational $n \times n$ matrix A with integer entries of bit length a, one can find an δ-forward approximation to its Jordan Normal Form $A = VJV^{-1}$ in time $\mathrm{poly}(n, a, \log(1/\delta))$, where the degree of the polynomial is at least 12. This

[6] This is called an *a fortiriori bound* in numerical analysis.

[7] [36] states: "A priori backward and forward error bounds for evaluation of the matrix sign function remain elusive."

[8] The output of their algorithm is n vectors on each of which Newton's method converges quadratically to an eigenvector, which they refer to as "approximation à la Smale".

algorithm works in the rational arithmetic model of computation, so it does not quite answer Demmel's question since it is not a numerically stable algorithm. However, it enjoys the significant advantage of being able to compute forward approximations to discontinuous quantities such as the Jordan structure.

As far as we are aware, there are no other published provably polynomial-time algorithms for the general eigenproblem. The two standard references for diagonalization appearing most often in theoretical computer science papers do not meet this criterion. In particular, the widely cited work by Pan and Chen [44] proves that one can compute the *eigenvalues* of A in $O(n^\omega + n\log\log(1/\delta))$ (suppressing logarithmic factors) *arithmetic* operations by finding the roots of its characteristic polynomial, which becomes a bound of $O(n^{\omega+1}a + n^2\log(1/\delta)\log\log(1/\delta))$ bit operations if the characteristic polynomial is computed exactly in rational arithmetic and the matrix has entries of bit length a. However that paper does not give any bound for the amount of time taken to find approximate eigenvectors from approximate eigenvalues, and states this as an open problem.[9]

Finally, the important work of Demmel, Dumitriu, and Holtz [12] (see also the followup [45]), which we rely on heavily, does not claim to provably solve the eigenproblem either—it bounds the running time of one iteration of a specific algorithm, and shows that such an iteration can be implemented numerically stably, without proving any bound on the number of iterations required in general.

Hermitian Eigenproblem. For comparison, the eigenproblem for Hermitian matrices is much better understood. We cannot give a complete bibliography of this huge area, but mention one relevant landmark result: the work of Wilkinson [46] and Hoffman-Parlett [47] in the 60's and 70's, which shows that the Hermitian eigenproblem can be solved with backward error δ in $O(n^3 + n^2\log(1/\delta))$ arithmetic operations with $O(\log(n/\delta))$ bits of precision. There has also recently been renewed interest in this problem in the theoretical computer science community, with the goal of bringing the runtime close to $O(n^\omega)$: Louis and Vempala [48] show how to find a $\delta-$approximation of just the largest eigenvalue in $O(n^\omega\log^4(n)\log^2(1/\delta))$ bit operations, and Ben-Or and Eldar [49] give an $O(n^{\omega+1}\mathrm{polylog}(n))$-bit-operation algorithm for finding a $1/\mathrm{poly}(n)$-approximate diagonalization of an $n\times n$ Hermitian matrix normalized to have $\|A\|\le 1$.

Remark I.9 (Davies' Conjecture)**.** The beautiful paper [50] introduced the idea of approximating a matrix function $f(A)$ for nonnormal A by $f(A+E)$ for some well-chosen E regularizing the eigenvectors of A. This directly inspired our approach to solving the eigenproblem via regularization.

The existence of an approximate diagonalization (1) for every A with a *well-conditioned similarity* V (i.e, $\kappa(V)$ depending polynomially on δ and n) was precisely the content of Davies' conjecture [50], which was recently solved by some of the authors and Mukherjee in [8]. The existence of such a V is a pre-requisite for proving that one can always efficiently find an approximate diagonalization in finite arithmetic, since if $\|V\|\|V^{-1}\|$ is very large it may require many bits of precision to represent. Thus, Theorem I.6 can be viewed as an efficient algorithmic answer to Davies' question.

II. Preliminaries

Let $M\in\mathbb{C}^{n\times n}$ be a complex matrix, not necessarily normal. We will write matrices and vectors with uppercase and lowercase letters, respectively. Let us denote by $\Lambda(M)$ the spectrum of M and by $\lambda_i(M)$ its individual eigenvalues. In the same way we denote the singular values of M by $\sigma_i(M)$ and we adopt the convention $\sigma_1(M)\ge\sigma_2(M)\ge\cdots\ge\sigma_n(M)$. When M is clear from the context we will simplify notation and just write Λ,λ_i or σ_i respectively.

Recall that the *operator norm* of M is

$$\|M\| = \sigma_1(M) = \sup_{\|x\|=1}\|Mx\|.$$

As usual, we will say that M is *diagonalizable* if it can be written as $M = VDV^{-1}$ for some diagonal matrix D whose nonzero entries contain the eigenvalues of M. In this case we have the spectral expansion

$$M = \sum_{i=1}^n \lambda_i v_i w_i^*, \tag{11}$$

where the right and left eigenvectors v_i and w_j^* are the columns and rows of V and V^{-1} respectively, normalized so that $w_i^* v_i = 1$.

A. Spectral Projectors and Holomorphic Functional Calculus

Let $M\in\mathbb{C}^{n\times n}$, with eigenvalues $\lambda_1,...,\lambda_n$. We say that a matrix P is a *spectral projector* for M if $MP = PM$ and $P^2 = P$. For instance, each of the terms $v_i w_i^*$ appearing in the spectral expansion (11) is a spectral projector, as $Av_iw_i^* = \lambda_i v_i w_i^* = v_i w_i^* A$ and $w_i^* v_i = 1$. If Γ_i is a simple closed positively oriented rectifiable curve in the complex plane separating λ_i from the rest of the spectrum, then it is well-known that

$$v_i w_i^* = \frac{1}{2\pi i}\oint_{\Gamma_i}(z-M)^{-1}\mathrm{d}z,$$

by taking the Jordan normal form of the the *resolvent* $(z-M)^{-1}$ and applying Cauchy's integral formula.

[9] "The remaining nontrivial problems are, of course, the estimation of the above output precision p [sufficient for finding an approximate eigenvector from an approximate eigenvalue], ... We leave these open problems as a challenge for the reader." – [44, Section 12].

Since every spectral projector P commutes with M, its range agrees exactly with an invariant subspace of M. We will often find it useful to choose some region of the complex plane bounded by a simple closed positively oriented rectifiable curve Γ, and compute the spectral projector onto the invariant subspace spanned by those eigenvectors whose eigenvalues lie inside Γ. Such a projector can be computed by a contour integral analogous to the above.

Recall that if f is any function, and M is diagonalizable, then we can meaningfully define $f(M) := Vf(D)V^{-1}$, where $f(D)$ is simply the result of applying f to each element of the diagonal matrix D. The *holomorphic functional calculus* gives an equivalent definition that extends to the case when M is non-diagonalizable. As we will see, it has the added benefit that bounds on the norm of the resolvent of M can be converted into bounds on the norm of $f(M)$.

Proposition II.1 (Holomorphic Functional Calculus). *Let M be any matrix, $B \supset \Lambda(M)$ be an open neighborhood of its spectrum (not necessarily connected), and $\Gamma_1, ..., \Gamma_k$ be simple closed positively oriented rectifiable curves in B whose interiors together contain all of $\Lambda(M)$. Then if f is holomorphic on B, the definition*

$$f(M) := \frac{1}{2\pi i}\sum_{j=1}^{k}\oint_{\Gamma_j} f(z)(z-M)^{-1}\mathrm{d}z$$

is an algebra homomorphism *in the sense that $(fg)(M) = f(M)g(M)$ for any f and g holomorphic on B.*

Finally, we will frequently use the *resolvent identity*

$$(z-M)^{-1}-(z-M')^{-1} = (z-M)^{-1}(M-M')(z-M')^{-1}$$

to analyze perturbations of contour integrals.

B. Pseudospectrum and Spectral Stability

The $\epsilon-$pseudospectrum of a matrix is defined in (6). Directly from this definition, we can relate the pseudospectra of a matrix and a perturbation of it.

Proposition II.2 ([19], Theorem 52.4). *For any $n \times n$ matrices M and E and any $\epsilon > 0$, $\Lambda_{\epsilon-\|E\|}(M) \subseteq \Lambda_\epsilon(M+E)$.*

It is also immediate that $\Lambda(M) \subset \Lambda_\epsilon(M)$, and in fact a stronger relationship holds as well:

Proposition II.3 ([19], Theorem 4.3). *For any $n \times n$ matrix M, any bounded connected component of $\Lambda_\epsilon(M)$ must contain an eigenvalue of M.*

Several other notions of stability will be useful to us as well. If M has distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, and spectral expansion as in (11), we define the *eigenvalue condition number of λ_i* to be

$$\kappa(\lambda_i) := \|v_i w_i^*\| = \|v_i\|\|w_i\|.$$

By considering the scaling of V in (2) in which its columns v_i have unit length, so that $\kappa(\lambda_i) = \|w_i\|$, we obtain the useful relationship

$$\kappa_V(M) \leq \|V\|\|V^{-1}\| \leq \|V\|_F\|V^{-1}\|_F \leq \sqrt{n \cdot \sum_{i\leq n}\kappa(\lambda_i)^2}. \tag{12}$$

Note also that the eigenvector condition number and pseudospectrum are related as follows:

Lemma II.4 ([19]). *Let $D(z,r)$ denote the open disk of radius r centered at $z \in \mathbb{C}$. For every $M \in \mathbb{C}^{n\times n}$,*

$$\bigcup_i D(\lambda_i, \epsilon) \subset \Lambda_\epsilon(M) \subset \bigcup_i D(\lambda_i, \epsilon\kappa_V(M)). \tag{13}$$

In this paper we will repeatedly use that assumptions about the pseudospectrum of a matrix can be turned into stability statements about functions applied to the matrix via the holomorphic functional calculus. Here we describe an instance of particular importance.

Let λ_i be a simple eigenvalue of M and let Γ_i be a contour in the complex plane, as in Section II-A, separating λ_i from the rest of the spectrum of M, and assume $\Lambda_\epsilon(M) \cap \Gamma = \emptyset$. Then, for any $\|M - M'\| < \eta < \epsilon$, a combination of Proposition II.2 and Proposition II.3 implies that there is a unique eigenvalue λ_i' of M' in the region enclosed by Γ, and furthermore $\Lambda_{\epsilon-\eta}(M') \cap \Gamma = \emptyset$. If v_i' and w_i' are the right and left eigenvectors of M' corresponding to λ_i' we have

$$\begin{aligned} \|v_i'w_i'^* - v_iw_i^*\| &= \frac{1}{2\pi}\left\|\oint_\Gamma (z-M)^{-1} - (z-M')^{-1}\mathrm{d}z\right\| \\ &= \frac{1}{2\pi}\left\|\oint_\Gamma (z-M)^{-1}(M-M')(z-M')^{-1}\mathrm{d}z\right\| \\ &\leq \frac{\ell(\Gamma)}{2\pi}\frac{\eta}{\epsilon(\epsilon-\eta)}. \end{aligned} \tag{14}$$

We have introduced enough tools to prove Proposition I.1.

Proof of Proposition I.1: For $t \in [0,1]$ define $A(t) = (1-t)A + tA'$. Since $\delta < \frac{\mathrm{gap}(A)}{8\kappa_V(A)}$ the Bauer-Fike theorem implies that $A(t)$ has distinct eigenvalues for all t, and in fact $\mathrm{gap}(A(t)) \geq \frac{3\mathrm{gap}(A)}{4}$. Standard results in perturbation theory [**?**] imply that for every $i = 1, \ldots, n$, $A(t)$ has a unique eigenvalue $\lambda_i(t)$ such that $\lambda_i(t)$ is a differentiable trajectory, $\lambda_i(0) = \lambda_i$ and $\lambda_i(1) = \lambda_i'$. Let $v_i(t)$ and $w_i(t)$ be the right and left eigenvectors of $\lambda_i(t)$ respectively, with $\|v_i(t)\| = 1$.

Let Γ_i be the positively oriented contour forming the boundary of the disk centered at λ_i with radius $\mathrm{gap}(A)/2$, and define $\epsilon = \frac{\mathrm{gap}(A)}{8\kappa_V(A)}$. Lemma II.4 implies $\Lambda_{2\epsilon}(A) \cap \Gamma_i = \emptyset$, and for fixed $t \in [0,1]$, since $\|A - A(t)\| < t\delta < \epsilon$, Proposition II.2 gives $\Lambda_\epsilon(A(t)) \cap \Gamma_i = \emptyset$. By (14)

$$|\kappa(\lambda_i) - \kappa(\lambda_i(t))| \leq \|v_i(t)w_i^*(t) - v_iw_i^*\| \leq \frac{\ell(\Gamma_i)\epsilon}{4\pi\epsilon^2} = 2\kappa_V(A),$$

and hence $\kappa(\lambda_i(t)) \le \kappa(\lambda_i) + 2\kappa_V(A) \le 3\kappa_V(A)$. Combining this with (12) we obtain

$$\kappa_V(A(t)) \le 2\sqrt{n \cdot \sum_i \kappa(\lambda_i)^2} < 4n\kappa_V(A).$$

On the other hand, from standard perturbation theory we know that the phases of the $v_i(t)$ may be chosen so that $v_i(t)$ is a differentiable function, and moreover one can show that

$$\|\dot{v}_i(t)\| \le \frac{\kappa_V(A(t))\|\dot{A}(t)\|}{\mathrm{gap}(A(t))} < \frac{\delta\kappa_V(A(t))}{\mathrm{gap}(A(t))};$$

see Section 2 of [**?**] or the references therein for a derivation of these facts. Now, using that $\kappa_V(A(t)) \le 4n\kappa_V(A)$ and $\mathrm{gap}(A(t)) \ge \frac{3\mathrm{gap}(A)}{4}$, the above inequality yields $\|\dot{v}_i(t)\| \le \frac{16n\delta\kappa_V(A)}{3\mathrm{gap}(A)}$. The desired result is then obtained by integrating $\dot{v}_i(t)$ from 0 to 1. ■

C. Finite-Precision Arithmetic

We briefly elaborate on the axioms for floating-point arithmetic given in Section I-A. Similar guarantees to the ones appearing in that section for scalar-scalar operations also hold for operations such as matrix-matrix addition and matrix-scalar multiplication. In particular, if A is an $n \times n$ complex matrix,

$$\mathsf{fl}(A) = A + A \circ \Delta \qquad |\Delta_{i,j}| < \mathbf{u}.$$

It will be convenient for us to write such errors in additive, as opposed to multiplicative form. We can convert the above to additive error as follows. Recall that for any $n \times n$ matrix, the spectral norm (the $\ell^2 \to \ell^2$ operator norm) is at most $\sqrt{n}$ times the $\ell^2 \to \ell^1$ operator norm, i.e. the maximal norm of a column. Thus we have

$$\|A \circ \Delta\| \le \sqrt{n} \max_i \|(A \circ \Delta)e_i\| \tag{15}$$

$$\le \sqrt{n} \max_{i,j} |\Delta_{i,j}| \max_i \|Ae_i\| \le \mathbf{u}\sqrt{n}\|A\|. \tag{16}$$

For more complicated operations such as matrix-matrix multiplication and matrix inversion, we use existing error guarantees from the literature. This is the subject of Section II-E.

We will also need to compute the trace of a matrix $A \in \mathbb{C}^{n\times n}$, and normalize a vector $x \in \mathbb{C}^n$. Error analysis of these is standard (see for instance the discussion in [16, Section 3.1-3.4, 4.1]) and the results in this paper are highly insensitive to the details. For simplicity, calling $\hat{x} := x/\|x\|$, we will assume that

$$|\mathsf{fl}\,(\mathrm{Tr}A) - \mathrm{Tr}A| \le n\|A\|\mathbf{u} \tag{17}$$

$$\|\mathsf{fl}(\hat{x}) - \hat{x}\| \le n\mathbf{u}. \tag{18}$$

Each of these can be achieved by assuming that $\mathbf{u}n \le \epsilon$ for some suitably chosen ϵ, independent of n, a requirement which will be depreciated shortly by several tighter assumptions on the machine precision.

Throughout the paper, we will take the pedagogical perspective that our algorithms are games played between the practitioner and an adversary who may additively corrupt each operation. In particular, we will include explicit error terms (always denoted by $E_{(\cdot)}$) in each appropriate step of every algorithm. In many cases we will first analyze a routine in exact arithmetic—in which case the error terms will all be set to zero—and subsequently determine the machine precision $\mathbf{u}$ necessary to make the errors small enough to guarantee convergence.

D. Sampling Gaussians in Finite Precision

For various parts of the algorithm, we will need to sample from normal distributions. For our model of arithmetic, we assume that the complex normal distribution can be sampled up to machine precision in $O(1)$ arithmetic operations. To be precise, we assume the existence of the following sampler:

Definition II.5 (Complex Gaussian Sampling)**.** A c_{N}-stable Gaussian sampler $\mathsf{N}(\sigma)$ takes as input $\sigma \in \mathbb{R}_{\ge 0}$ and outputs a sample of a random variable $\widetilde{G} = \mathsf{N}(\sigma)$ with the property that there exists $G \sim N_{\mathbb{C}}(0, \sigma^2)$ satisfying

$$|\widetilde{G} - G| \le c_{\mathsf{N}}\sigma \cdot \mathbf{u}$$

with probability one, in at most T_{N} arithmetic operations for some universal constant $T_{\mathsf{N}} > 0$. We will only sample $O(n^2)$ Gaussians during the algorithm, so this sampling will not contribute significantly to the runtime. Here as everywhere in the paper, we will omit issues of underflow or overflow. Throughout this paper, to simplify some of our bounds, we will also assume that $c_{\mathsf{N}} \ge 1$.

E. Black-box Error Assumptions for Multiplication, Inversion, and QR

Our algorithm uses matrix-matrix multiplication, matrix inversion, and QR factorization as primitives. For our analysis, we must therefore assume some bounds on the error and runtime costs incurred by these subroutines. In this section, we first formally state the kind of error and runtime bounds we require, and then discuss some implementations known in the literature that satisfy each of our requirements with modest constants.

Our definitions are inspired by the definition of *logarithmic stability* introduced in [12]. Roughly speaking, they say that implementing the algorithm with floating point precision $\mathbf{u}$ yields an accuracy which is at most polynomially or quasipolynomially in n worse than $\mathbf{u}$ (possibly also depending on the condition number in the case of inversion). Their definition has the property that while a logarithmically stable algorithm is not strictly-speaking backward stable, it can attain the same forward error bound as a backward stable algorithm at the cost of increasing the bit length by a polylogarithmic factor. See Section 3 of their paper for a precise definition and a more detailed discussion of how their definition relates to standard numerical stability notions.

Definition II.6. A $\mu_{\mathsf{MM}}(n)$*-stable multiplication algorithm* $\mathsf{MM}(\cdot,\cdot)$ takes as input $A, B \in \mathbb{C}^{n\times n}$ and a precision $\mathbf{u} > 0$ and outputs $C = \mathsf{MM}(A,B)$ satisfying

$$\|C - AB\| \le \mu_{\mathsf{MM}}(n) \cdot \mathbf{u}\|A\|\|B\|,$$

on a floating point machine with precision $\mathbf{u}$, in $T_{\mathsf{MM}}(n)$ arithmetic operations.

Definition II.7. A $(\mu_{\mathsf{INV}}(n), c_{\mathsf{INV}})-$*stable inversion algorithm* $\mathsf{INV}(\cdot)$ takes as input $A \in \mathbb{C}^{n\times n}$ and a precision $\mathbf{u}$ and outputs $C = \mathsf{INV}(A)$ satisfying

$$\|C - A^{-1}\| \le \mu_{\mathsf{INV}}(n) \cdot \mathbf{u} \cdot \kappa(A)^{c_{\mathsf{INV}} \log n}\|A^{-1}\|,$$

on a floating point machine with precision $\mathbf{u}$, in $T_{\mathsf{INV}}(n)$ arithmetic operations.

Definition II.8. A $\mu_{\mathsf{QR}}(n)$*-stable QR factorization algorithm* $\mathsf{QR}(\cdot)$ takes as input $A \in \mathbb{C}^{n\times n}$ and a precision $\mathbf{u}$, and outputs $[Q, R] = \mathsf{QR}(A)$ such that

1) R is exactly upper triangular.
2) There is a unitary Q' and a matrix A' such that

$$Q'A' = R, \tag{19}$$

and

$$\|Q' - Q\| \le \mu_{\mathsf{QR}}(n)\mathbf{u}, \quad \text{and} \quad \|A' - A\| \le \mu_{\mathsf{QR}}(n)\mathbf{u}\|A\|,$$

on a floating point machine with precision $\mathbf{u}$. Its running time is $T_{\mathsf{QR}}(n)$ arithmetic operations.

Remark II.9. Throughout this paper, to simplify some of our bounds, we will assume that

$$1 \le \mu_{\mathsf{MM}}(n), \mu_{\mathsf{INV}}(n), \mu_{\mathsf{QR}}(n), c_{\mathsf{INV}} \log n.$$

The above definitions can be instantiated with traditional $O(n^3)$-complexity algorithms for which $\mu_{\mathsf{MM}}, \mu_{\mathsf{QR}}, \mu_{\mathsf{INV}}$ are all $O(n)$ and $c_{\mathsf{INV}} = 1$ [16]. This yields easily-implementable practical algorithms with running times depending cubically on n.

In order to achieve $O(n^\omega)$-type efficiency, we instantiate them with fast-matrix-multiplication-based algorithms and with $\mu(n)$ taken to be a low-degree polynomial [12]. Specifically, the following parameters are known to be achievable.

Theorem II.10 (Fast and Stable Instantiations of $\mathsf{MM}, \mathsf{INV}, \mathsf{QR}$)**.**

1) *If ω is the exponent of matrix multiplication, then for every $\eta > 0$ there is a $\mu_{\mathsf{MM}}(n)-$stable multiplication algorithm with $\mu_{\mathsf{MM}}(n) = n^{c_\eta}$ and $T_{\mathsf{MM}}(n) = O(n^{\omega+\eta})$, where c_η does not depend on n.*
2) *Given an algorithm for matrix multiplication satisfying (1), there is a $(\mu_{\mathsf{INV}}(n), c_{\mathsf{INV}})$-stable inversion algorithm with*

$$\mu_{\mathsf{INV}}(n) \le O(\mu_{\mathsf{MM}}(n) n^{\lg(10)}), \qquad c_{\mathsf{INV}} \le 8,$$

and $T_{\mathsf{INV}}(n) \le T_{\mathsf{MM}}(3n) = O(T_{\mathsf{MM}}(n))$.
3) *Given an algorithm for matrix multiplication satisfying (1), there is a $\mu_{\mathsf{QR}}(n)-$stable QR factorization algorithm with*

$$\mu_{\mathsf{QR}}(n) = O(n^{c_{\mathsf{QR}}} \mu_{\mathsf{MM}}(n)),$$

where c_{QR} is an absolute constant, and $T_{\mathsf{QR}}(n) = O(T_{\mathsf{MM}}(n))$.

In particular, all of the running times above are bounded by $T_{\mathsf{MM}}(n)$ for an $n \times n$ matrix.

Proof: (1) is Theorem 3.3 of [1]. (2) is Theorem 3.3 (see also equation (9) above its statement) of [12]. The final claim follows by noting that $T_{\mathsf{MM}}(3n) = O(T_{\mathsf{MM}}(n))$ by dividing a $3n \times 3n$ matrix into nine $n \times n$ blocks and proceeding blockwise, at the cost of a factor of 9 in $\mu_{\mathsf{INV}}(n)$. (3) appears in Section 4.1 of [12]. ■

We remark that for specific existing fast matrix multiplication algorithms such as Strassen's algorithm, specific small values of $\mu_{\mathsf{MM}}(n)$ are known (see [1] and its references for details), so these may also be used as a black box, though we will not do this in this paper.

III. Full Version

The full version of this paper with all proofs and details referenced above is available at [14].

Acknowledgments

We thank Peter Bürgisser for introducing us to this problem, and Ming Gu, Olga Holtz, Vishesh Jain, Ravi Kannan, Pravesh Kothari, Lin Lin, Satyaki Mukherjee, Yuji Nakatsukasa, and Nick Trefethen for helpful conversations. We also thank the Institute for Pure and Applied Mathematics, where part of this work was carried out.

J.B. is supported by the NSF Graduate Research Fellowship Program under Grant DGE-1752814. N.S. is supported by NSF Grant CCF-1553751.

References

[1] J. Demmel, I. Dumitriu, O. Holtz, and R. Kleinberg, "Fast matrix multiplication is stable," *Numerische Mathematik*, vol. 106, no. 2, pp. 199–224, 2007.

[2] A. N. Beavers Jr. and E. D. Denman, "A new similarity transformation method for eigenvalues and eigenvectors," *Mathematical Biosciences*, vol. 21, no. 1-2, pp. 143–169, 1974.

[3] D. Armentano, C. Beltrán, P. Bürgisser, F. Cucker, and M. Shub, "A stable, polynomial-time algorithm for the eigenpair problem," *Journal of the European Mathematical Society*, vol. 20, no. 6, pp. 1375–1437, 2018.

[4] B. N. Parlett, *The symmetric eigenvalue problem.* SIAM, 1998, vol. 20.

[5] J. D. Roberts, "Linear model reduction and solution of the algebraic Riccati equation by use of the sign function," *International Journal of Control*, vol. 32, no. 4, pp. 677–687, 1980.

[6] R. Byers, "Numerical stability and instability in matrix sign function based algorithms," in *Computational and Combinatorial Methods in Systems Theory*. Citeseer, 1986.

[7] J.-y. Cai, "Computing Jordan normal forms exactly for commuting matrices in polynomial time," *International Journal of Foundations of Computer Science*, vol. 5, no. 03n04, pp. 293–302, 1994.

[8] J. Banks, A. Kulkarni, S. Mukherjee, and N. Srivastava, "Gaussian regularization of the pseudospectrum and Davies' conjecture," *arXiv preprint arXiv:1906.11819, to appear in Communications on Pure and Applied Mathematics*, 2019.

[9] D. Shi and Y. Jiang, "Smallest gaps between eigenvalues of random matrices with complex Ginibre, Wishart and universal unitary ensembles," *arXiv preprint arXiv:1207.4240*, 2012.

[10] S. Ge, "The eigenvalue spacing of iid random matrices and related least singular value results," Ph.D. dissertation, UCLA, 2017.

[11] C. S. Kenney and A. J. Laub, "The matrix sign function," *IEEE Transactions on Automatic Control*, vol. 40, no. 8, pp. 1330–1348, 1995.

[12] J. Demmel, I. Dumitriu, and O. Holtz, "Fast linear algebra is stable," *Numerische Mathematik*, vol. 108, no. 1, pp. 59–91, 2007.

[13] J. W. Demmel, "On condition numbers and the distance to the nearest ill-posed problem," *Numerische Mathematik*, vol. 51, no. 3, pp. 251–289, 1987.

[14] J. Banks, J. G. Vargas, A. Kulkarni, and N. Srivastava, "Pseudospectral shattering, the sign function, and diagonalization in nearly matrix multiplication time," *arXiv preprint arXiv:1912.08805*, 2019.

[15] J. W. Demmel, *Applied numerical linear algebra*. SIAM, 1997, vol. 56.

[16] N. J. Higham, *Accuracy and stability of numerical algorithms*. SIAM, 2002, vol. 80.

[17] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*. Springer Science & Business Media, 2012, vol. 2.

[18] S. Smale, "Complexity theory and numerical analysis," *Acta Numerica*, vol. 6, pp. 523–551, 1997.

[19] L. N. Trefethen and M. Embree, *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005.

[20] Z. Bai and J. Demmel, "Using the matrix sign function to compute invariant subspaces," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 1, pp. 205–225, 1998.

[21] J. P. Vinson, "Closest spacing of eigenvalues," *arXiv preprint arXiv:1111.2743*, 2011.

[22] G. B. Arous and P. Bourgade, "Extreme gaps between eigenvalues of random matrices," *The Annals of Probability*, vol. 41, no. 4, pp. 2648–2681, 2013.

[23] H. Nguyen, T. Tao, and V. Vu, "Random matrices: tail bounds for gaps between eigenvalues," *Probability Theory and Related Fields*, vol. 167, no. 3-4, pp. 777–816, 2017.

[24] M. Aizenman, R. Peled, J. Schenker, M. Shamis, and S. Sodin, "Matrix regularizing effects of Gaussian perturbations," *Communications in Contemporary Mathematics*, vol. 19, no. 03, p. 1750028, 2017.

[25] J. Von Neumann and H. H. Goldstine, "Numerical inverting of matrices of high order," *Bulletin of the American Mathematical Society*, vol. 53, no. 11, pp. 1021–1099, 1947.

[26] S. Smale, "On the efficiency of algorithms of analysis," *Bulletin (New Series) of The American Mathematical Society*, vol. 13, no. 2, pp. 87–121, 1985.

[27] J. W. Demmel, "The probability that a numerical analysis problem is difficult," *Mathematics of Computation*, vol. 50, no. 182, pp. 449–480, 1988.

[28] A. Edelman, "Eigenvalues and condition numbers of random matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, no. 4, pp. 543–560, 1988.

[29] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM (JACM)*, vol. 51, no. 3, pp. 385–463, 2004.

[30] A. Sankar, D. A. Spielman, and S.-H. Teng, "Smoothed analysis of the condition numbers and growth factors of matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 2, pp. 446–476, 2006.

[31] P. Śniady, "Random regularization of Brown spectral measure," *Journal of Functional Analysis*, vol. 193, no. 2, pp. 291–313, 2002.

[32] U. Haagerup and F. Larsen, "Brown's spectral distribution measure for R-diagonal elements in finite von Neumann algebras," *Journal of Functional Analysis*, vol. 176, no. 2, pp. 331–367, 2000.

[33] A. N. Beavers and E. D. Denman, "A computational method for eigenvalues and eigenvectors of a matrix with real eigenvalues," *Numerische Mathematik*, vol. 21, no. 5, pp. 389–396, 1973.

[34] E. D. Denman and A. N. Beavers Jr., "The matrix sign function and computations in systems," *Applied mathematics and Computation*, vol. 2, no. 1, pp. 63–94, 1976.

[35] A. N. Malyshev, "Parallel algorithm for solving some spectral problems of linear algebra," *Linear algebra and its applications*, vol. 188, pp. 489–520, 1993.

[36] R. Byers, C. He, and V. Mehrmann, “The matrix sign function method and the computation of invariant subspaces,” *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 3, pp. 615–632, 1997.

[37] Z. Bai, J. Demmel, and M. Gu, “An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems,” *Numerische Mathematik*, vol. 76, no. 3, pp. 279–308, 1997.

[38] N. J. Higham, *Functions of matrices: theory and computation*. SIAM, 2008, vol. 104.

[39] ——, “The matrix sign decomposition and its relation to the polar decomposition,” *Linear Algebra and its Applications*, vol. 212, pp. 3–20, 1994.

[40] R. Byers and H. Xu, “A new scaling for Newton’s iteration for the polar decomposition and its backward stability,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 822–843, 2008.

[41] Y. Nakatsukasa and N. J. Higham, “Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD,” *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. A1325–A1349, 2013.

[42] Y. Nakatsukasa and R. W. Freund, “Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev’s functions,” *SIAM Review*, vol. 58, no. 3, pp. 461–493, 2016.

[43] D. Bindel, S. Chandresekaran, J. Demmel, D. Garmire, and M. Gu, “A fast and stable nonsymmetric eigensolver for certain structured matrices,” Technical report, University of California, Berkeley, CA, Tech. Rep., 2005.

[44] V. Y. Pan and Z. Q. Chen, “The complexity of the matrix eigenproblem,” in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. ACM, 1999, pp. 507–516.

[45] G. Ballard, J. Demmel, and I. Dumitriu, “Minimizing communication for eigenproblems and the singular value decomposition,” *arXiv preprint arXiv:1011.3077*, 2010.

[46] J. H. Wilkinson, “Global convergence of tridiagonal QR algorithm with origin shifts,” *Linear Algebra and its Applications*, vol. 1, no. 3, pp. 409–420, 1968.

[47] W. Hoffmann and B. N. Parlett, “A new proof of global convergence for the tridiagonal QL algorithm,” *SIAM Journal on Numerical Analysis*, vol. 15, no. 5, pp. 929–937, 1978.

[48] A. Louis and S. S. Vempala, “Accelerated newton iteration for roots of black box polynomials,” in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2016, pp. 732–740.

[49] M. Ben-Or and L. Eldar, “A quasi-random approach to matrix spectral analysis,” in *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[50] E. B. Davies, “Approximate diagonalization,” *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 4, pp. 1051–1064, 2007.