Invariant subspaces and PCA in nearly matrix multiplication time

Aleksandros Sobczyk
IBM Research and ETH Zurich
obc@zurich.ibm.com

Marko Mladenović
ETH Zurich
mmladenovic@ethz.ch

Mathieu Luisier
ETH Zurich
mluisier@iis.ee.ethz.ch

Abstract

Approximating invariant subspaces of generalized eigenvalue problems (GEPs) is a fundamental computational problem at the core of machine learning and scientific computing. It is, for example, the root of Principal Component Analysis (PCA) for dimensionality reduction, data visualization, and noise filtering, and of Density Functional Theory (DFT), arguably the most popular method to calculate the electronic structure of materials. Given Hermitian $\mathbf{H}, \mathbf{S} \in \mathbb{C}^{n \times n}$, where $\mathbf{S}$ is positive-definite, let $\mathbf{\Pi}_k$ be the true spectral projector on the invariant subspace that is associated with the k smallest (or largest) eigenvalues of the GEP $\mathbf{HC} = \mathbf{SC\Lambda}$, for some $k \in [n]$. We show that we can compute a matrix $\widetilde{\mathbf{\Pi}}_k$ such that $\|\mathbf{\Pi}_k - \widetilde{\mathbf{\Pi}}_k\|_2 \le \epsilon$, in $O\left(n^{\omega+\eta} \operatorname{polylog}(n, \epsilon^{-1}, \kappa(\mathbf{S}), \operatorname{gap}_k^{-1})\right)$ bit operations in the floating point model, for some $\epsilon \in (0, 1)$, with probability $1 - 1/n$. Here, $\eta > 0$ is arbitrarily small, $\omega \lesssim 2.372$ is the matrix multiplication exponent, $\kappa(\mathbf{S}) = \|\mathbf{S}\|_2 \|\mathbf{S}^{-1}\|_2$, and gap_k is the gap between eigenvalues k and $k+1$. To achieve such provable "forward-error" guarantees, our methods rely on a new $O(n^{\omega+\eta})$ stability analysis for the Cholesky factorization, and a smoothed analysis for computing spectral gaps, which can be of independent interest. Ultimately, we obtain new matrix multiplication-type bit complexity upper bounds for PCA problems, including classical PCA and (randomized) low-rank approximation.

1 Introduction

Generalized eigenvalue problems (GEPs) arise naturally in a plethora of applications in machine learning, scientific computing, and engineering. Given a pair of matrices $\mathbf{H}$ and $\mathbf{S}$, often referred to as a *matrix pencil*, the problem of interest has the following form

$$\mathbf{HC} = \mathbf{SC\Lambda}, \tag{1}$$

where $\mathbf{C}$ and $\mathbf{\Lambda}$ are the unknown eigenvector and eigenvalue matrices, respectively. Of particular importance are the so-called "Hermitian definite" or simply "definite" GEPs/pencils, in which case $\mathbf{H}$ is Hermitian and $\mathbf{S}$ is Hermitian and positive-definite. In many important applications, the quantity of interest is an (arbitrarily large) subset of eigenvectors, defining an *invariant subspace*, rather than the entire $\mathbf{C}$ and $\mathbf{\Lambda}$ solutions of the GEP.

In data science and machine learning, invariant subspaces play a central role in many problems, including Spectral Clustering [111, 124], Language Models [73], Image Processing [118, 11], Recommendation Systems [46], Principal Components Analysis (PCA) [45, 79, 122], Support Vector Machines [99], and many others [50, 35, 96, 12]. We particularly focus on PCA applications, which can take the form of a GEP as in Eq. (1) where $\mathbf{H}$ is the sample covariance and $\mathbf{S}$ the identity. In more advanced settings, $\mathbf{H}$ and $\mathbf{S}$ can be defined over a kernel [122, 20]; See Section 4 and Appendix G for more details. Another closely related application comes from Density Functional

Theory [88] (DFT), which is not a machine learning problem per se, but it is probably the most commonly used method (it was awarded the Nobel prize in Chemistry in 1998) to compute the electronic and structural properties of materials. In this case, $\mathbf{H}$ is the Hamiltonian and $\mathbf{S}$ the overlap matrix (cf. Appendix F). The spectral projector on the invariant subspace corresponding to the smallest generalized eigenvalues (occupied energies) directly provides the density matrix and the electron density. Obtaining them often presents a challenge from the computational point of view.

1.1 Problem definition

The main focus of this work is the computation of *spectral projectors* on invariant subspaces that are associated with a subset of the spectrum of Hermitian definite GEPs. As the Abel-Ruffini theorem excludes exact computation, even in exact arithmetic, we seek for approximate computations, as described in the following Problem 1.1.

Problem 1.1 (Spectral projector). *Given a Hermitian definite GEP* $\mathbf{HC} = \mathbf{SC\Lambda}$ *of size* n, *an integer* $1 \leq k \leq n-1$, *and accuracy* $\epsilon \in (0,1)$, *compute a matrix* $\widetilde{\mathbf{\Pi}}_k \in \mathbb{C}^{n \times n}$ *such that*

$$\left\| \widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k \right\| \leq \epsilon, \tag{2}$$

where $\mathbf{\Pi}_k$ *is the true spectral projector on the invariant subspace associated with the* k *smallest or largest eigenvalues.*

Before proposing algorithms to solve Problem 1.1, we first make some clarifications and define useful concepts.

Type of approximation: The approximation of the form of Equation (2) is commonly called a "forward approximation" or "forward error" in numerical analysis. It quantifies the distance between the true solution of the problem and the one returned by an algorithm. It is a stronger and harder to achieve notion of approximation than the related "backward error." For details see Appendix A.1.

Model of computation: While many finite precision models of computation exist in the literature, all algorithms in this work are analyzed in the floating point model of computation, which is also the prominent model implemented in existing computers. Each real number α is rounded to a floating point number $\mathbf{fl}(\alpha) = (1 + \theta)\alpha$, where $\theta \in \mathbb{C}$ satisfies $|\theta| \leq \mathbf{u} \in \mathbb{R}_{>0}$. The machine precision $\mathbf{u}$ bounds also the errors introduced by arithmetic operations $\{+, -, \times, /\}$, and the expression $\log(1/\mathbf{u})$ gives the number of bits required to achieve the desired precision. More details can be found in Appendix A.2.

Bit complexity: The complexity of numerical algorithms is often measured in terms of the arithmetic operations executed, commonly referred to as *arithmetic complexity*. A more realistic notion is the *bit complexity*, which bounds the number of boolean operations. In the floating point model, it is straightforward to translate the arithmetic to the bit complexity if we have an upper bound on the number of bits. For instance, arithmetic operations on b bits can be typically carried out in $O(b^2)$ bit operations, or even faster by using more advanced algorithms [123, 55, 68].

Matrix multiplication time: In two seminal works [39, 40], it was demonstrated that matrix multiplication and other fundamental problems in Numerical Linear Algebra can be solved in the floating point model with nearly $O(n^{\omega+\eta})$ bit-complexity (up to polylogarithmic factors), where η is an arbitrarily small positive number and ω is the matrix multiplication exponent, to-date bounded by $\omega \lesssim 2.372$ [47, 141, 6]. Hereafter, we will use the notation $T_{\mathsf{MM}}(n) = n^{\omega+\eta}$.

1.2 Existing algorithms

Here we give a brief overview of existing algorithms. We refer to Appendix A.7 for more details. GEPs in general can be solved using classic eigensolvers and related techniques in $\widetilde{O}(n^3)$ floating operations, e.g., by reducing the matrix (or pencil) to tridiagonal form with similarity transformations and applying the shifted QR algorithm on the tridiagonal matrix, or by using a divide-and-conquer method (see [37, 109, 75, 8, 9, 64, 42, 10, 113] and references therein). Significant progresses beyond the $\widetilde{O}(n^3)$ bit complexity barrier have been made [40, 39, 15, 95, 25, 41, 107, 121].

Regarding the computation of eigenvalues, two notable examples are the $\widetilde{O}(T_{\mathsf{MM}}(n))$ algorithm of [95] for the largest eigenvalye, and the $\widetilde{O}(n^2)$ algorithm of [107] for the spectral norm.

The first to have addressed the problem of computing invariant subspaces in nearly $O(T_{\mathsf{MM}}(n))$ in floating point is [39] (see also [14, 13]). The authors described an iterative algorithm for the Schur decomposition, and showed that each individual step is numerically stable, and it takes $O(T_{\mathsf{MM}}(n))$ operations. An end-to-end bound on the number of iterations to achieve a backward approximate solution was left open. More recently, the seminal work of [15] extended the analysis to obtain an end-to-end $\widetilde{O}(T_{\mathsf{MM}}(n))$ complexity to approximately diagonalize a matrix, and [41, 121] provided a rigorous analysis for the generalized eigenproblem case. In Corollary 1.7 and Proposition 1.1 of [15], it was also outlined how to translate the backward diagonalization error to a forward error for the eigenvectors. The reported bound, however, has two main limitations: it relies on simplicity of the spectrum, which is a strict assumption, and it requires as input an over-estimate on the eigenvector condition number of the problem, which is unknown, and [15] does not describe how to compute it (see Appendix B.1 for more details). In this work we describe how to overcome these limitations and provide a novel, end-to-end, provably accurate analysis (in the sense of Eq. (2)) for arbitrary invariant subspaces of definite GEPs with $\widetilde{O}(T_{\mathsf{MM}}(n))$ boolean complexity.

1.3 Contributions and methods

Our main contribution, summarized in the following Theorem 1.1 and Algorithm 1, is an end-to-end analysis to solve Problem 1.1 in nearly $O(T_{\mathsf{MM}}(n))$ time.

Theorem 1.1. *Let* $(\mathbf{H}, \mathbf{S})$ *be a Hermitian definite pencil of size* n *with* $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\| \leq 1$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ *its eigenvalues,* $\mathrm{gap}_k = \lambda_{k+1} - \lambda_k$ *and* $\kappa(\mathbf{S}) = \|\mathbf{S}\|\|\mathbf{S}^{-1}\|$. *Algorithm 1*

$$\widetilde{\mathbf{\Pi}}_k \leftarrow \mathsf{PROJECTOR}(\mathbf{H}, \mathbf{S}, k, \epsilon),$$

takes as inputs $\mathbf{H}$, $\mathbf{S}$, *an integer* $k \in [n-1]$, *an error parameter* $\epsilon \in (0, 1)$ *and returns a matrix* $\widetilde{\mathbf{\Pi}}_k$ *such that*

$$\Pr\left[\left\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\right\| \leq \epsilon\right] \geq 1 - 1/n,$$

where $\mathbf{\Pi}_k$ *is the true spectral projector on the invariant subspace that is associated with the* k *smallest (or largest) eigenvalues. The algorithm executes*

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}) + \log(n\kappa(\mathbf{S}))\log(\kappa(\mathbf{S})) + \log\left(\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)\right)\right)$$

floating point operations with $O\left(\log(n)\left(\log^4(\tfrac{n}{\mathrm{gap}_k}) + \log^4(n\kappa(\mathbf{S})) + \log^3(\tfrac{1}{\epsilon\,\mathrm{gap}_k})\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)\right)$ *bits of precision. Internally, the algorithm needs to generate a total of at most* $\widetilde{O}(n)$ *standard normal floating point numbers using additional* $O(\log(\log(n)))$ *bits.*

To achieve the results of Theorem 1.1, we provide a novel $O(T_{\mathsf{MM}}(n))$-type complexity analysis of several problems in numerical linear algebra that can be of independent interest.

In brief, our methodology is as follows. We first observe that if we can determine reasonable "guesses" for the spectral gap $(\widetilde{\mathrm{gap}}_k)$ and for the midpoint $(\widetilde{\mu}_k)$ between the λ_k and λ_{k+1} eigenvalues then we can efficiently compute the spectral projector by approximating the sign function

$$\mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}),$$

using the analysis of [15] for the Newton iteration. The matrix $\frac{1}{2}(\mathbf{I} + \mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}))$ indeed transforms in exact arithmetic all eigenvalues that are smaller than $\widetilde{\mu}_k$ to 1 and the ones larger than $\widetilde{\mu}_k$ to zero. As will be proved in Proposition 2.1, in Section 2, this approach is sufficient to provide an accurate spectral projector $\widetilde{\mathbf{\Pi}}_k$ in floating point. As a consequence, the problem reduces to approximating the aforementioned midpoint and gap. As a baseline, in Appendix B.1 we prove that this can be done in nearly $O(T_{\mathsf{MM}}(n))$ with iterative inversion [39] and diagonalization [15] or, similarly, by iteratively calling generalized diagonalization [41]. However, this approach presents two drawbacks: It does not take advantage of the inherent symmetry of the problem, and, at the same time, it performs a full diagonalization when we are only interested in the gap between two

specific eigenvalues, which is seemingly redundant. We formally prove this claim by designing a novel approach that achieves better complexity, typically by a factor of $O(\log(n))$ (cf. Section 3.4). Importantly, no explicit diagonalization is necessary.

To minimize the complexity of our algorithm, it is crucial to leverage symmetry. To that end we use the Cholesky factorization of $\mathbf{S}$ in the spirit of the Cholesky-QR algorithm [37]. We highlight that, while other factorizations have been solved in $O(T_{\mathsf{MM}}(n))$ in floating point, an end-to-end analysis for Cholesky remains open. In exact arithmetic, for example, the LU of a Hermitian definite matrix directly provides its Cholesky and [39] showed that the LU factorization of non-symmetric matrices can be obtained in $O(T_{\mathsf{MM}}(n))$. However, when considering arithmetic errors, the relationship between LU and Cholesky does not hold in floating point, as demonstrated by the counter-example of Appendix C.5. Other fast Cholesky algorithms have been proposed for special classes of matrices, e.g., for matrices with well-defined separators [58, 94, 61] and graph Laplacians [91, 92]. However, they do not generalize to arbitrary dense matrices. Our analysis is the first to improve the classic $O(n^3)$ floating point Cholesky algorithms [84, 71] for the general case, with provable error bounds. In the following Theorem 1.2 we summarize our new analysis of the Cholesky factorization Algorithm 2 (see also Appendix C). We note that the algorithm itself is not new, only its analysis.

Theorem 1.2. *Given a Hermitian positive-definite matrix $\mathbf{M}$, there exists an algorithm $\mathbf{L} \leftarrow$ CHOLESKY$(\mathbf{M})$, listed in Algorithm 2, which requires $O(T_{\mathsf{MM}}(n))$ arithmetic operations. This algorithm is logarithmically stable, in a sense that, there exist global constants c_1, c_2, c_3, such that for all $\epsilon \in (0, 1)$, if executed in a floating point machine with precision*

$$\mathbf{u} \leq \mathbf{u}_{\mathsf{CHOLESKY}} := \epsilon \frac{1}{c_1 n^{c_2} \kappa(\mathbf{M})^{c_3 \log n}},$$

which translates into $O\left(\log(n) \log(\kappa(\mathbf{M})) + \log(\frac{1}{\epsilon})\right)$ required bits of precision, then it does not break down due to arithmetic errors, and the solution returned satisfies $\|\mathbf{L}\mathbf{L}^* - \mathbf{M}\| \leq \epsilon \|\mathbf{M}\|$.

This stand-alone result fulfills the definition of "logarithmic-stability," a notion of numerical stability that is commonly used in the related literature [39, 15]. Given this new Cholesky analysis, the following transformation of the GEP to a regular Hermitian eigenvalue problem:

$$\mathbf{H}\mathbf{C} = \mathbf{S}\mathbf{C}\mathbf{\Lambda} \Rightarrow \mathbf{L}^*\mathbf{H}\mathbf{L}(\mathbf{L}^{-1}\mathbf{C}) = (\mathbf{L}^{-1}\mathbf{C})\mathbf{\Lambda},$$

can be carried out accurately in $O(T_{\mathsf{MM}}(n))$ in floating point, with provable forward-error bounds for all eigenvalues of the transformed problem. Here, $\mathbf{L}$ is the Cholesky factor of $\mathbf{S}^{-1}$ instead of $\mathbf{S}$. Specifically, in Proposition C.3 in Appendix C.4, we prove that the corresponding Algorithm 4, $\widetilde{\mathbf{H}} \leftarrow$ REDUCE$(\mathbf{H}, \mathbf{S}, \epsilon)$, returns a Hermitian matrix $\widetilde{\mathbf{H}}$ such that, for any given accuracy $\epsilon \in (0, 1)$,

$$\left|\lambda_i(\widetilde{\mathbf{H}}) - \lambda_i(\mathbf{S}^{-1}\mathbf{H})\right| \leq \epsilon, \forall i \in [n].$$

The symmetry induced by the Cholesky transformation is crucial to design an efficient algorithm for the spectral gap. As described in Section 3, and analyzed in Appendices D and E, any spectral gap or eigenvalue of a Hermitian definite pencil can be approximated by an iterative algorithm that uses only "counting-queries", i.e., queries that ask how many eigenvalues are smaller than a given threshold. This way we completely avoid diagonalization, thus leading to a lower complexity.

To perform the counting queries efficiently, the transformed matrix $\widetilde{\mathbf{H}}$ must be regularized with small random perturbations, in the spirit of smoothed analysis [128], which has recently drawn attention in the context of matrix algorithms [31, 25, 15, 41, 104] (see Appendix D for the analysis). These aforementioned works typically require a guarantee on the minimum eigenvalue gap of the perturbed matrix, e.g., [104] uses a Minami-type bound [105], while in [15, 41] the entire pseudospectrum of the perturbed matrix must be shattered with respect to a grid. The latter is even more challenging to achieve than a minimum gap and it requires $\widetilde{O}(n^2)$ random bits. Our algorithm is significantly less demanding in terms of randomness: All we need is the Wegner estimate [138, 4] for the density-of-states of random Hermitian operators, and only $\widetilde{O}(n)$ random bits in total.

Finally, in Section 4, we apply our main results to prove the first matrix multiplication-type upper bounds for the bit complexity of PCA algorithms. Specifically, for the standard PCA formulation, we show that we can first compute the spectral projector and then use deflation to obtain a basis for the desired low-dimensional embedding in nearly matrix multiplication time. We then apply similar arguments to the seminal Block-Krylov PCA algorithm of [106].

<div style="border:1px solid black; padding:10px;">

<div align="center">PROJECTOR</div>

Input: Hermitian definite pencil $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}^n_{++}$, gap index k, accuracy ϵ.

Requires: $\|\mathbf{H}\| \leq 1$, $\|\mathbf{S}^{-1}\| \leq 1$, $k \in [n-1]$.

Algorithm: $\widetilde{\mathbf{\Pi}} \leftarrow \mathsf{PROJECTOR}(\mathbf{H}, \mathbf{S}, k, \epsilon)$.

1: $\widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k \leftarrow \mathsf{GAP}(\mathbf{H}, \mathbf{S}, k, \frac{1}{8}, \frac{1}{2n})$.

2: $\widetilde{\kappa} \leftarrow \mathsf{COND}(\mathbf{S}, \frac{1}{4}, \frac{1}{2n})$.   ▷ This is skipped when $\mathbf{S} = \mathbf{I}$.

3: $\widetilde{\mathbf{\Pi}} \leftarrow \mathsf{PURIFY}(\mathbf{H}, \mathbf{S}, \widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k, \widetilde{\kappa}, \epsilon)$.

4: **return** $\widetilde{\mathbf{\Pi}}$.

Output: Approximate projector $\widetilde{\mathbf{\Pi}}$ on the invariant subspace associated with the k smallest eigenvalues.

Ensures: $\|\widetilde{\mathbf{\Pi}} - \mathbf{\Pi}\| \leq \epsilon$ with probability at least $1 - 1/n$.

</div>

<div align="center">**Algorithm 1:** $\mathsf{PROJECTOR}$.</div>

<div style="border:1px solid black; padding:10px;">

<div align="center">CHOLESKY</div>

Input: Matrix $\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \in \mathbb{H}^n_{++}$.

Requires: $\mathbf{M}$ is positive-definite, $\mathbf{u} \leq \epsilon \frac{1}{c_1 n^{c_2} \kappa(\mathbf{M})^{c_3 \log(n)}}$ for some constants c_1, c_2, c_3 and $\epsilon \in (0, 1)$.

Algorithm: $\mathbf{L} \leftarrow \mathsf{CHOLESKY}(\mathbf{M})$.

1: **if** $n = 1$ **:**

2:     **return** $\mathbf{L} = \sqrt{\mathbf{M}_{1,1}}$.

3: **else**

4:     $\mathbf{L}_{11} \leftarrow \mathsf{CHOLESKY}(\mathbf{A})$.   ▷ $\mathbf{L}_{11}\mathbf{L}_{11}^* = \mathbf{A} + \mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}$.

5:     $\mathbf{BA}i \leftarrow \mathsf{MM}(\mathbf{B}, \mathsf{INV}(\mathbf{A}))$.   ▷ $\mathbf{BA}i = \mathbf{B}\left(\mathbf{A}^{-1} + \mathbf{E}_1^{\mathsf{INV}}\right) + \mathbf{E}_2^{\mathsf{MM}} = \mathbf{BA}^{-1} + \mathbf{E}_{\mathbf{BA}i}$.

6:     $\mathbf{L}_{21} \leftarrow \mathsf{MM}(\mathbf{BA}i, \mathbf{L}_{11})$.   ▷ $\mathbf{L}_{21} = \left(\mathbf{BA}^{-1} + \mathbf{E}_{\mathbf{BA}i}\right)\mathbf{L}_{11} + \mathbf{E}_3^{\mathsf{MM}} = \mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{E}_{\mathbf{L}_{21}}$.

7:     $\widetilde{\mathbf{S}} \leftarrow \mathbf{C} - \mathsf{HERM}(\mathsf{MM}(\mathbf{BA}i, \mathbf{B}^*))$.   ▷ $\widetilde{\mathbf{S}} = \mathbf{C} - \left((\mathbf{BA}i)\mathbf{B}^* + \mathbf{E}_4^{\mathsf{MM}}\right) + \mathbf{E}_5^{\mathsf{SUB}} = \mathbf{S} + \mathbf{E}_{\mathbf{S}}$,

        ▷ where $\mathbf{E}_{\mathbf{S}} = \mathbf{BE}_1^{\mathsf{INV}}\mathbf{B}^* + \mathbf{E}_2^{\mathsf{MM}}\mathbf{B}^* + \mathbf{E}_4^{\mathsf{MM}} + \mathbf{E}_5^{\mathsf{SUB}}$.

8:     $\mathbf{L}_{22} \leftarrow \mathsf{CHOLESKY}(\widetilde{\mathbf{S}})$.   ▷ $\mathbf{L}_{22}\mathbf{L}_{22}^* = \widetilde{\mathbf{S}} + \mathbf{E}_{\widetilde{\mathbf{S}}}^{\mathsf{CH}}$.

9:     **return** $\mathbf{L} = \begin{pmatrix} \mathbf{L}_{11} & \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix}$.

Output: Lower triangular Cholesky factor $\mathbf{L} \in \mathbb{C}^{n \times n}$.

Ensures: $\|\mathbf{LL}^* - \mathbf{M}\| \leq \epsilon\|\mathbf{M}\|$, $\mathbf{L}$ is lower triangular, and $\mathbf{LL}^*$ is Hermitian and positive-definite.

</div>

<div align="center">**Algorithm 2:** $\mathsf{CHOLESKY}(\mathbf{M})$.</div>

1.4 Notation

Matrices are denoted by bold capital letters and vectors by bold small letters. For real or complex constants we typically use Greek letters, or the Latin letters c, C. The vector $\mathbf{e}_i$ denotes the i-th column of the standard basis. $\mathbf{A}^*$ is the conjugate transpose of $\mathbf{A}$ and $\mathbf{A}^\dagger$ denotes the pseudoinverse. The 2-norm is the default for matrices and vectors. $\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^\dagger\|$ is the two-norm condition number of $\mathbf{A}$. For the error analysis of the various algorithms, we use $\mathbf{E}_i^{\mathsf{OP}}$ to denote the error matrices that are introduced by the floating point errors of the i-th operation OP. The letters ϵ and δ typically denote (scalar) error quantities and failure probabilities, respectively. $[n]$ is the set $\{1, 2, ..., n\}$. We denote by $\mathbb{H}^n \subset \mathbb{R}^n$ the set of Hermitian matrices of size $n \times n$, $\mathbb{H}^n_+$ the set of Hermitian positive semi-definite matrices and $\mathbb{H}^n_{++}$ the set of Hermitian positive definite matrices. For a matrix $\mathbf{A}$ and a scalar z we write $z \pm \mathbf{A}$ as a shorthand for $z\mathbf{I} \pm \mathbf{A}$. $\Lambda(\mathbf{A})$ and $\Lambda(\mathbf{A}, \mathbf{B})$ denote the spectrum of a matrix $\mathbf{A}$ and a matrix pencil $(\mathbf{A}, \mathbf{B})$, respectively. The eigenvalues and

singular values are always sorted in ascending order by default: $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$. $\Lambda_\epsilon(\mathbf{A})$ is the ϵ-pseudospectrum of $\mathbf{A}$ (see Definition A.2).

2 Computing spectral projectors with the sign function

Given a Hermitian definite pencil $(\mathbf{H}, \mathbf{S})$, our ultimate goal is to compute a forward error approximation of the spectral projector that is associated with the k smallest eigenvalues, as described in Problem 1.1. Algorithm 3 solves this problem provably and efficiently, but it requires that we already have a suitable approximation of the eigenvalue gap that separates the desired subspace from the rest of the eigenspace. The algorithm is called PURIFY since it is inspired by "purification" techniques in DFT, referring to the removal of the unoccupied orbitals. The computation of the gap and the midpoint is in fact the bottleneck of our main algorithm, however, we still show that they can be computed efficiently in Section 3, and, importantly, without diagonalizing any matrices. The properties of Algorithm 3 are stated in Proposition 2.1.

Proposition 2.1. *Let* $\mathbf{H} \in \mathbb{H}^n$ *with* $\|\mathbf{H}\| \le 1$, $\mathbf{S} \in \mathbb{H}^n_{++}$ *with* $\|\mathbf{S}^{-1}\| \le 1$, $k \in [n-1]$ *and* $\epsilon \in (0,1)$. *Let* $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$ *and* $\mathrm{gap}_k = \lambda_k - \lambda_{k+1}$, *where* $\lambda_1 \le \ldots \le \lambda_n$ *are the generalized eigenvalues of the Hermitian definite pencil* $(\mathbf{H}, \mathbf{S})$ *and assume that we want to compute* $\mathbf{\Pi}_k$ *which is the true spectral projector associated with the* k *smallest eigenvalues. If we have access to*

$$\widetilde{\mu}_k \in \mu_k \pm \tfrac{1}{8}\,\mathrm{gap}_k \quad \widetilde{\mathrm{gap}}_k \in (1 \pm \tfrac{1}{8})\,\mathrm{gap}_k, \quad \widetilde{\kappa} \in [\kappa(\mathbf{S}), C\kappa(\mathbf{S})],$$

for some constant $C > 1$, *then Algorithm 3 computes* $\widetilde{\mathbf{\Pi}}_k \leftarrow \mathsf{PURIFY}(\mathbf{H}, \mathbf{S}, \widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k, \widetilde{\kappa}, \epsilon)$ *such that* $\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\| \le \epsilon$, *in* $O\left(T_{\mathsf{MM}}(n)\left(\log(\frac{1}{\mathrm{gap}_k}) + \log(\log(\frac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k}))\right)\right)$ *floating point operations using* $O\left(\log(n)\log^3(\frac{1}{\mathrm{gap}_k})\log(\frac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)$ *bits of precision.*

Proof. The full proof of Proposition 2.1 can be found in Appendix B. We briefly summarize it here. The main idea is to use the sign function algorithm from [15], SGN, to approximate $\mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})$. If we already know that $\widetilde{\mu}_k$ is a reasonable approximation of μ_k, and that it is located inside the correct eigenvalue gap, then, in exact arithmetic, our problem is equivalent to computing $\mathrm{sgn}(\mu_k - \mathbf{S}^{-1}\mathbf{H})$. The result can be used to filter the desired spectral projector, often referred as "purification" in the context of DFT. The main challenge is to ensure that all propagated numerical errors, success probabilities, and input parameters for all algorithms are well-defined and bounded. To obtain the final forward errors we must rely on matrix similarity arguments, the properties of the pseudospectrum, the eigenvalue bounds of Weyl and Kahan from Fact A.1, and Lemma B.1, which gives explicit bounds on the sign function under small floating point perturbations. $\qquad\square$

The rest of the paper is devoted to the analysis of our new algorithm for the spectral gap and the midpoint based on eigenvalue counting queries, described in Theorem 3.1. For comparison purposes, in Appendix B.1 we analyze a diagonalization-based algorithm for the same task (which is a new result itself), specifically, using the state-of-the-art EIG algorithm of [15]. We compare the two algorithms in Section 3.4, demonstrating that our counting-based algorithm is indeed faster.

3 Fast spectral gaps with counting queries

Our core algorithm efficiently approximates spectral gaps based on "eigenvalue counting queries" only, thus avoiding an explicit (and expensive) diagonalization. To give some intuition on the main idea, consider the following simplified version of the problem.

Problem 3.1 (Gap finder). *Let* $\lambda_1 \le \ldots \le \lambda_n$ *in* $[-1,1]$ *be* n *(unknown) real values (e.g., they can be the eigenvalues of the original matrix pencil)* $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$, *and* $\mathrm{gap}_k = \lambda_{k+1} - \lambda_k$, *for some* $k \in [n-1]$. *Given* k *and some error parameter* $\epsilon \in (0, 1/2)$ *as input, we want to approximate* μ_k *and* gap_k *up to additive* $\epsilon\,\mathrm{gap}_k$, *i.e., we look for* $\widetilde{\mu}_k = \mu_k \pm \epsilon\,\mathrm{gap}_k$ *and* $\widetilde{\mathrm{gap}}_k \in (1 \pm \epsilon)\,\mathrm{gap}_k$. *Only queries of the following form can be performed: We fix a parameter* $\gamma \in (0, 1/2)$, *which distorts all* λ_i *to some (unknown)* $\lambda'_i \in [\lambda_i - \gamma, \lambda_i + \gamma]$. *We can then choose any value* $h \in [-1-\gamma, 1+\gamma]$ *and ask how many values* λ'_i *are smaller than* h. *For each* γ, *we can query arbitrarily many different values for* h, *and each* h-*query costs* $q(1/\gamma) = O(\mathrm{polylog}(1/\gamma))$.

The query cost is arbitrary to avoid trivial solutions by setting $\gamma = 0$. The following proposition is proved in Appendix E:

Proposition 3.1. *Problem 3.1 can be solved iteratively by executing a total of $O(\log(\frac{1}{\epsilon \, \text{gap}_k}))$ iterations and $\Theta(1)$ queries per iteration, where each query costs at most $q(\frac{1}{\epsilon \, \text{gap}_k})$.*

3.1 Smoothed analysis of eigenvalue counting

To use the counting query model of Problem 3.1 and Proposition 3.1 to compute the spectral gap of a matrix pencil, we need a "black-box" method to count eigenvalues that are smaller than a threshold. We first describe a straightforward, deterministic algorithm $\textsf{COUNT}(\widetilde{\mathbf{X}}, h, \varepsilon)$ for this task (see Lemma E.1), which takes as input a Hermitian matrix $\widetilde{\mathbf{X}}$, a scalar h, and a parameter ε, with the requirement that $\sigma_{\min}(h - \widetilde{\mathbf{X}}) > \varepsilon$. It returns the precise number of eigenvalues of $\widetilde{\mathbf{X}}$ that are smaller than h. The runtime of the algorithm depends on $\log(1/\varepsilon)$, and must therefore be minimized. For this we resort to smoothed analysis: We apply a random perturbation to ensure that ε is at least polynomial in $1/n$, up to some other factors detailed in Appendix D.

To build a random "regularizer," in Definition D.1 we introduce a random oracle that samples numbers from a standard normal distribution and returns their floating point representation using a pre-specified number of bits. Based on this simple oracle, we can design a floating point algorithm $\widetilde{\mathbf{X}} \leftarrow \textsf{REGULARIZE}(\mathbf{A}, \gamma, \delta)$ which has the following properties described in Proposition 3.2:

Proposition 3.2. *Let $\mathbf{A}$ with $\|\mathbf{A}\| \leq 1$ be a Hermitian matrix, $\gamma, \delta \in (0, 1/4)$ two given parameters, and $\widetilde{\mathbf{X}} \leftarrow \textsf{REGULARIZE}(\mathbf{A}, \gamma, \delta)$. Let $\textsf{g}$ be an arbitrary (but fixed) grid of points in $[-2, 2]$ with cardinality $|\textsf{g}| = T$. For every element $h_i \in \textsf{g}$ consider the matrices $\mathbf{M}_i = h_i - \widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{M}} = h_i - \widetilde{\mathbf{X}} + \mathbf{E}_i$, where $\mathbf{E}_i$ denote the diagonal floating point error matrices induced by the shift. All the following hold simultaneously with probability $1 - 2\delta$ if we use $O(\log(\frac{Tn}{\gamma \delta}))$ bits of precision:*

$$\|\widetilde{\mathbf{X}}\| \leq 4/3, \quad \left|\lambda_i(\widetilde{\mathbf{X}}) - \lambda_i(\mathbf{A})\right| \leq \tfrac{9}{16}\gamma, \quad \sigma_{\min}(\widetilde{\mathbf{M}}_i) \geq \frac{\gamma \delta}{4nT\sqrt{4\pi \ln(4n/\delta)}}.$$

Proof. The main result that we use in the proof can be traced back to the Wegner estimate for the density-of-states of Hermitian operators under random diagonal disorder [138]. See Appendix D and in particular D.2 for more details. $\square$

3.2 Computing the gap and the midpoint

We can now describe the algorithm GAP in Theorem 3.1, that computes the k-th gap and the midpoint of a Hermitian definite pencil. The same methodology can be extended to approximate any singular value, as described in Proposition E.2 in Appendix E.3.

Theorem 3.1 (GAP). *Let $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}^n_{++}$ and $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\| \leq 1$, which define a Hermitian definite pencil $(\mathbf{H}, \mathbf{S})$. Given $k \in [n-1]$, accuracy $\epsilon \in (0, 1)$, and failure probability $\delta \in (0, 1/2)$, there exists an algorithm*

$$\widetilde{\mu}_k, \widetilde{\text{gap}}_k \leftarrow \textsf{GAP}(\mathbf{H}, \mathbf{S}, k, \epsilon, \delta)$$

which returns $\widetilde{\mu}_k = \mu_k \pm \epsilon \, \text{gap}_k$ and $\widetilde{\text{gap}}_k = (1 \pm \epsilon)\,\text{gap}_k$, where $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$ and $\text{gap}_k = \lambda_k - \lambda_{k+1}$. The algorithm requires

$$O\left(T_{\textsf{MM}}(n)\log(\tfrac{1}{\delta \epsilon \, \text{gap}_k})\log(\tfrac{1}{\epsilon \, \text{gap}_k})\right)$$

arithmetic operations using $O\left(\log(n)\left(\log^4(\frac{n}{\delta \epsilon \, \text{gap}_k}) + \log(\kappa(\mathbf{S}))\right)\right)$ bits, where λ_i are the eigenvalues of $(\mathbf{H}, \mathbf{S})$. If $\kappa(\mathbf{S})$ is unknown, additional $O(T_{\textsf{MM}}(n)\log(\frac{n\kappa(\mathbf{S})}{\delta})\log(\kappa(\mathbf{S})))$ floating point operations and $O(\log(n)\log^4(\frac{n\kappa(\mathbf{S})}{\delta}))$ bits are sufficient to compute it with Corollary E.1.

Proof. The full proof builds upon the results that are detailed in Appendices D and E. A summary is the following. We first fix our initial error parameter $\gamma_0 = 1/8$ and call $\widetilde{\mathbf{H}} = \textsf{REDUCE}\left(\mathbf{H}, \mathbf{S}, \frac{\gamma_0}{4}\right)$

7

(Algorithm 4), which internally uses CHOLESKY to reduce the GEP to a regular Hermitian one. From Proposition C.3, a Hermitian matrix $\widetilde{\mathbf{H}}$ is returned such that $|\lambda_i(\widetilde{\mathbf{H}}) - \lambda_i(\mathbf{H}, \mathbf{S})| \leq \frac{\gamma_0}{4}$.

Next, we use the same counting query model as in Proposition 3.1. We first regularize $\widetilde{\mathbf{H}}$ using $\widetilde{\mathbf{X}} \leftarrow \mathsf{REGULARIZE}(\widetilde{\mathbf{H}}, \frac{\gamma_0}{2}, \delta_0)$, where $\delta_0 = \delta/2$ is the initial failure probability. Conditioning on success of Proposition 3.2 (with probability $1 - \delta_0$), for all $i \in [n]$, it holds that $|\lambda_i(\widetilde{\mathbf{X}}) - \lambda_i(\widetilde{\mathbf{H}})| \leq 9\gamma_0/16$. Summing the two eigenvalue error bounds, we conclude that all eigenvalues of $(\mathbf{H}, \mathbf{S})$, which initially lie in $[-1, 1]$, are distorted by at most γ_0 in $\widetilde{\mathbf{X}}$. We now have all necessary tools to go back to the counting query model of Proposition 3.1: In the first step we construct a grid $\mathsf{g} = \{-1, -7/8, -6/8, \ldots, 7/8, 1, 9/8\}$. Clearly, $|\mathsf{g}| = \Theta(1)$. Since we conditioned on the success of Proposition 3.2, the regularization ensures that for every $h_j \in \mathsf{g}$ it holds that $\sigma_{\min}(h_j - \widetilde{\mathbf{X}} + \mathbf{E}) \geq \varepsilon_0$ with $\varepsilon_0 = \frac{\gamma_0 \delta_0}{8|\mathsf{g}|n\sqrt{\pi \ln(4n/\delta_0)}}$. This allows us to efficiently execute $\mathsf{COUNT}(\widetilde{\mathbf{X}}, h_j, \varepsilon_0)$ for every h_j.

At the end of the first iteration, we have computed two intervals I_k and I_{k+1}, where I_k contains λ_k and I_{k+1} contains λ_{k+1}, and each interval has size at most 1, i.e., half the size of $[-1, 1]$. We continue by halving at each step both γ and δ, constructing the corresponding grids as per the proof of Proposition 3.1, and counting eigenvalues over the grid. In each iteration after the first one, we keep track of two intervals I_k and I_{k+1}, and two corresponding grids g_k and g_{k+1} with size $|\mathsf{g}_k| = |\mathsf{g}_{k+1}| = \Theta(1)$. We therefore only need to execute a constant number of COUNT queries, and in each iteration the size of the intervals I_k and I_{k+1} is halved. The algorithm terminates after a total of $m = O\left(\log(\frac{1}{\epsilon \, \mathrm{gap}_k})\right)$ iterations and finally provides the advertised complexity, bit requirements, failure probability, and approximation guarantees. $\qquad\square$

3.3 Sketch proof of Theorem 1.1

The proof of our main Theorem 1.1 directly follows from Theorem 3.1 together with Proposition 2.1 as well as the algorithm SIGMAK (described in Appendix E.3) which is used to compute the condition number of $\mathbf{S}$. The full proof can be found in Appendix E.4.

3.4 Comparison with diagonalization

We can now compare Theorem 3.1 with a diagonalization-based approach that is detailed in Proposition B.2. We fix $\delta = O(1/n)$ so that both algorithms succeed with the same probability.

For $\epsilon, \mathrm{gap}_k, \kappa^{-1}(\mathbf{S}) \in \Omega(\mathrm{poly}(1/n))$, the total arithmetic complexity of the algorithm of Theorem 3.1 is $O(T_{\mathsf{MM}}(n)\log^2(n))$ using $O(\log^5(n))$ bits. For the same parameters, Proposition B.2 requires need a total of $O(T_{\mathsf{MM}}(n)\log^3(n))$ arithmetic operations, and $O(\log^5(n))$ bits. Thus, in total, Algorithm 3.1 is faster by a factor of $O(\log(n))$.

In the extreme case where $\epsilon, \mathrm{gap}_k, \kappa(\mathbf{S}) = \Theta(1)$, Theorem 3.1 counts $O(T_{\mathsf{MM}}(n)\log(n))$ arithmetic operations and $O(\log^5(n))$ bits, while Proposition B.2 requires $O(T_{\mathsf{MM}}(n)\log^2(n))$ operations, and $O(\log^5(n))$ bits. Thus, Proposition B.2 is again slower by a factor of $O(\log(n))$. Interestingly, in this case Theorem 3.1 is faster than even a single call to EIG, which requires $O(T_{\mathsf{MM}}(n)\log^2(n))$ arithmetic operations. We conclude that, at least based on the currently existing algorithms, diagonalization is redundant for the computation of spectral gaps and invariant subspaces.

3.5 Application in DFT

In Appendix F we demonstrate how our main results can be directly applied to approximate density matrices and electron densities of atomic systems in DFT. Even though is not a machine learning problem per se, we decided to dedicate a section in the Appendix due to its importance: DFT calculations persistently occupy supercomputing clusters and the corresponding software libraries and literature receive tens of thousands of citations annually at the time of this writing [89, 59, 60, 127, 74]. Our work is the first analysis to provide forward-error guarantees in finite precision for these problems in nearly matrix multiplication time.

4 PCA

Since its introduction in the early twentieth century [116, 72], Principal Component Analysis is one of the most important tools in statistics, data science, and machine learning. It can be used, for example, to visualize data, to reduce dimensionality, or to remove noise from data; cf. [79, 45] for reviews on the vast bibliography. In its simplest formulation, given a (centered) data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the goal is to learn a k-dimensional embedding $\mathbf{C}_k$, where $k < n$, that maximizes the sample variance, which can be written as an optimization problem

$$\mathbf{C}_k = \arg \max_{\mathbf{C}^\top \mathbf{C} = \mathbf{I}_{k \times k}} \operatorname{tr}(\mathbf{C}^\top \mathbf{H} \mathbf{C}), \tag{3}$$

where $\mathbf{H} = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{n \times n}$ is the sample covariance. It can be shown that the solution $\mathbf{C}_k$ corresponds to the principal k singular vectors of $\mathbf{H}$, i.e. the ones that correspond to the largest k singular values. Evidently, since the sample covariance is always symmetric and positive semi-definite, this can be written as a Hermitian eigenvalue problem

$$\mathbf{H}\mathbf{C} = \mathbf{C}\boldsymbol{\Lambda},$$

(which is indeed a definite GEP as in Equation (1) with $\mathbf{S} = \mathbf{I}$). By solving for $\mathbf{C}_k$, we can project the data in k dimensions by computing $\mathbf{X}\mathbf{C}_k$, preserving as much of the variance in k dimensions as possible. To compute $\mathbf{C}_k$ we can directly use our main results. However, the solution of Equation (3) is an actual orthonormal basis for the invariant subspace rather than the spectral projector that Theorem 1.1 returns. This can be addressed with deflation: Once we have the spectral projector $\widetilde{\boldsymbol{\Pi}}_k$, assuming that the approximation is sufficiently tight, we can apply a subsequent deflation step based on a rank-revealing QR factorization to obtain a k-dimensional basis. This can be done deterministically in $O(n^3)$ time [66] or in randomized $O(n^\omega)$ [39, 15].

In Appendix G.1 we prove the following Theorem 4.1 for Algorithm 7, which builds upon our main Theorem 1.1, the algorithm of Proposition E.2 to approximate $\|\mathbf{X} - \mathbf{X}_k\|$, and the DEFLATE algorithm of [19], to solve the standard PCA problem of Eq. (3). Following the existing literature, the result is stated for real matrices, but it can be trivially adapted to the complex case as well.

Theorem 4.1 (PCA). *Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a centered data matrix, $\mathbf{H}$ the $n \times n$ symmetric sample covariance matrix, i.e., $\mathbf{H} = \mathbf{X}^\top \mathbf{X}$, $\|\mathbf{H}\| \leq 1$, $k \in [n]$ a target rank, and $\epsilon \in (0,1)$ an accuracy parameter. Given $\mathbf{H}$, we can compute a matrix $\widetilde{\mathbf{C}}_k$ with k columns such that $\|\mathbf{X} - \mathbf{X}\widetilde{\mathbf{C}}_k \widetilde{\mathbf{C}}_k^\top\| \leq (1 + \epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{C}_k \mathbf{C}_k^\top\|$, where $\mathbf{C}_k \in \mathbb{R}^{n \times k}$ contains the top-k (right) singular vectors of $\mathbf{X}$ in*

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\sigma_{k+1}})\log(\tfrac{1}{\sigma_{k+1}}) + \log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}) + \log(\log(\tfrac{n}{\epsilon\sigma_{k+1}\,\mathrm{gap}_k}))\right)\right)$$

arithmetic operations using $O\left(\log(n)\left(\log^4(\tfrac{n}{\epsilon\,\mathrm{gap}_k}) + \log^4(\tfrac{n}{\sigma_{k+1}})\right) + \log(\tfrac{1}{\epsilon\sigma_{k+1}})\right)$ bits of precision, with probability at least $1 - O(1/n)$.

4.1 Block-Krylov PCA

In some applications, the target dimension k might be small, i.e., $k \ll n$. This condition has driven a whole area of research in so-called low-rank approximation algorithms for PCA [53, 120, 33, 67, 100, 106, 30, 29, 5]. Such approaches are also suitable for kernel PCA, since they rely on matrix-vector products and therefore the kernel matrix does not need to be explicitly formed. The techniques from the previous section can be directly applied to obtain new bit complexity upper bounds for existing state-of-the-art algorithms, which are typically analyzed in exact arithmetic. They internally rely on the computation of the principal singular vectors of submatrices, which can be improved with our methods. Specifically, in Appendix G.2 we summarize a floating point analysis of the Block-Krylov Iteration algorithm (see Algorithm 8), essentially, providing a matrix multiplication-type upper bound on the bit complexity with only a polylogarithmic dependence on the singular value gap. In a nutshell, we directly obtain the following result:

Theorem 4.2 (Bit complexity analysis of Block-Krylov PCA). *Let $\mathbf{X}$ be a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\|\mathbf{X}\| \leq 1$, $k \in [n]$ a target rank, $\epsilon_{\mathsf{PCA}} \in (0,1)$ an accuracy parameter, and $q = \Theta\left(\tfrac{\log(n)}{\sqrt{\epsilon_{\mathsf{PCA}}}}\right)$. Let $T_{\mathsf{MMX}}(k)$ denote the complexity to stably multiply $\mathbf{X}$ or $\mathbf{X}^\top$ with a dense matrix with k columns*

from the right (see Def. G.1). Using the Steps 1-6 that are detailed in Appendix G.3 as a floating point implementation of Algorithm 8, we can compute a matrix $\widetilde{\mathbf{Z}}_k \in \mathbb{R}^{m \times k}$ that satisfies

$$\left\| \widetilde{\mathbf{Z}}_k \widetilde{\mathbf{Z}}_k^\top - \mathbf{Z}_k \mathbf{Z}_k^\top \right\| \le O(\epsilon_{\mathsf{PCA}}),$$

with high probability, where $\mathbf{Z}_k$ is an approximate basis for the top-k principal components of $\mathbf{X}$, returned by Algorithm 8 in exact arithmetic. The total cost is at most

$$O\left(qT_{\mathsf{MMX}}(k) \log\left(\tfrac{\kappa(\mathbf{K})}{\mathrm{gap}_k(\mathbf{M})}\right) + m(qk)^{\omega-1} \log\left(\tfrac{1}{\mathrm{gap}_k(\mathbf{M})}\right) + (qk)^\omega \operatorname{polylog}\left(\tfrac{qk}{\mathrm{gap}_k(\mathbf{M})}\right) \right)$$

floating point operations, using $O\left(\operatorname{polylog}\left(\tfrac{mq\kappa(\mathbf{K})}{\epsilon_{\mathsf{PCA}}\, \mathrm{gap}_k}\right) \right)$ bits of precision. $\mathbf{K}, \mathbf{M}$ are as in Alg. 8.

Proof. The full proof can be found in Thm. G.2, Appendix G.3. The main idea is to apply the counting query methodology to compute the condition number of the Block-Krylov matrix $\mathbf{K}$, as well as the k-th spectral gap and the midpoint of the reduced matrix $\mathbf{M}$ in Line 5 of Alg. 8. Thereafter, we can compute a spectral projector and an approximate basis for the top-k singular vectors of $\mathbf{M}$ using PURIFY and DEFLATE, similar to the analysis of classical PCA in the previous section. □

5   Open problems

We mention some open problems and interesting future directions.

(i) **Bit requirement of** SGN**:** The major bottleneck for the bit requirements of our main algorithms comes from the SGN algorithm of [15]. An inverse-free Newton-Schultz iteration [83], or the implicit repeated squaring of [41, 121] can potentially give significant improvements.

(ii) **Sparse algorithms:** In applications like DFT it commonly occurs that the matrices have special structure, i.e., they are banded and/or sparse. It remains open whether Problem 1.1 can be provably solved faster than our reported results in finite precision for these special cases (recall that the tridiagonal QR algorithm requires $O(n^3)$ operations to return the eigenvectors). An end-to-end stability analysis of existing fast eigensolvers would be the place to start [65, 137].

(iii) **Distributed PCA:** The techniques for Block-Krylov PCA can be potentially applied to distributed or streaming PCA algorithms, which are also based on randomized low-rank approximations. E.g., in the distributed PCA algorithm of [31], it is straightforward to replace the SVD computation on the server by a counting query iteration. The full analysis of such an approach is left as future work.

6   Conclusion

In this work we provided an end-to-end analysis to approximate spectral projectors on k-dimensional invariant subspaces of Hermitian definite matrix pencils $(\mathbf{H}, \mathbf{S})$ that require at most $O\left(T_{\mathsf{MM}}(n) \operatorname{polylog}(n, \epsilon^{-1}, \kappa(\mathbf{S}), \mathrm{gap}_k^{-1}) \right)$ bit operations in the floating point model of computation. This is the first end-to-end analysis that improves the $\widetilde{O}(n^3)$ complexity of classic eigensolvers for both the regular and the generalized case. To achieve this result we introduced a new method to approximate spectral gaps by querying the number of eigenvalues that are smaller than a threshold, and therefore completely avoid an explicit diagonalization of any matrix or pencil. This approach required proving that the Cholesky factorization can be stably computed in $O(T_{\mathsf{MM}}(n))$ floating point operations, a novel result *per se*. Our results have direct implications on PCA problems, providing matrix multiplication type upper bounds for the bit complexity of classical and Block-Krylov PCA.

Acknowledgements

References

[1] Dimitris Achlioptas. Database-friendly random projections. In *Proc. Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 274–281, 2001. 61

[2] Satoru Adachi, Satoru Iwata, Yuji Nakatsukasa, and Akiko Takeda. Solving the trust-region subproblem by a generalized eigenvalue problem. *SIAM Journal on Optimization*, 27(1):269–291, 2017. 26

[3] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proc. Thirty-Eighth Annual ACM Symposium on Theory of Computing*, pages 557–563, 2006. 62

[4] Michael Aizenman, Ron Peled, Jeffrey Schenker, Mira Shamis, and Sasha Sodin. Matrix regularizing effects of Gaussian perturbations. *Communications in Contemporary Mathematics*, 19(03):1750028, 2017. 4, 47

[5] Zeyuan Allen-Zhu and Yuanzhi Li. LazySVD: Even faster SVD decomposition yet without agonizing pain. *Advances in Neural Information Processing Systems*, 29, 2016. 9

[6] Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. *arXiv preprint arXiv:2404.16349*, 2024. 2

[7] Diego Armentano, Carlos Beltrán, Peter Bürgisser, Felipe Cucker, and Michael Shub. A stable, polynomial-time algorithm for the eigenpair problem. *Journal of the European Mathematical Society*, 20(6):1375–1437, 2018. 26

[8] Zhaojun Bai and James Demmel. *On a direct algorithm for computing invariant subspaces with specified eigenvalues*. University of Tennessee, Computer Science Department, 1991. 2, 26

[9] Zhaojun Bai and James Demmel. Using the matrix sign function to compute invariant subspaces. *SIAM Journal on Matrix Analysis and Applications*, 19(1):205–225, 1998. 2, 26, 27

[10] Zhaojun Bai, James Demmel, and Ming Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numerische Mathematik*, 76(3):279–308, 1997. 2, 26

[11] Gökhan H Bakır, Jason Weston, and Bernhard Schölkopf. Learning to find pre-images. *Advances in Neural Information Processing Systems*, 16:449–456, 2004. 1

[12] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989. 1

[13] Grey Ballard, James Demmel, and Ioana Dumitriu. Minimizing Communication for Eigenproblems and the Singular Value Decomposition. *Technical Report UCB/EECS-2011-14*, February 2011. 3

[14] Grey Ballard, James Demmel, Olga Holtz, and Oded Schwartz. Minimizing Communication in Numerical Linear Algebra. *SIAM Journal on Matrix Analysis and Applications*, 32(3):866–901, 2011. 3

[15] Jess Banks, Jorge Garza-Vargas, Archit Kulkarni, and Nikhil Srivastava. Pseudospectral Shattering, the Sign Function, and Diagonalization in Nearly Matrix Multiplication Time. *Foundations of Computational Mathematics*, pages 1–89, 2022. 2, 3, 4, 6, 9, 10, 19, 21, 22, 23, 26, 27, 31, 32, 33

[16] Jess Banks, Jorge Garza-Vargas, and Nikhil Srivastava. Global Convergence of Hessenberg Shifted QR II: Numerical Stability. *arXiv preprint arXiv:2205.06810*, 2022. 26

[17] Jess Banks, Jorge Garza-Vargas, and Nikhil Srivastava. Global Convergence of Hessenberg Shifted QR III: Approximate Ritz Values via Shifted Inverse Iteration. *arXiv preprint arXiv:2205.06804*, 2022. 26

[18] Jess Banks, Jorge Garza-Vargas, and Nikhil Srivastava. Global Convergence of Hessenberg Shifted QR I: Exact Arithmetic. *Foundations of Computational Mathematics*, pages 1–34, 2024. 26

[19] Jess Banks, Archit Kulkarni, Satyaki Mukherjee, and Nikhil Srivastava. Gaussian Regularization of the Pseudospectrum and Davies' Conjecture. *Communications on Pure and Applied Mathematics*, 74(10):2114–2131, 2021. 9, 62

[20] Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011. 1

[21] Yair Bartal, Ben Recht, and Leonard J Schulman. Dimensionality reduction: beyond the Johnson-Lindenstrauss bound. In *Proc. 2011 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 868–887. SIAM, 2011. 62

[22] Harold Basch, C. J. Hornback, and J. W. Moskowitz. Gaussian-Orbital Basis Sets for the First-Row Transition-Metal Atoms. *The Journal of Chemical Physics*, 51(4):1311–1318, 1969. 58

[23] Robert A Baston and Yuji Nakatsukasa. Stochastic diagonal estimation: probabilistic bounds and an improved algorithm. *arXiv preprint arXiv:2201.10684*, 2022. 62

[24] Friedrich L Bauer and Charles T Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2:137–141, 1960. 20

[25] Michael Ben-Or and Lior Eldar. A Quasi-Random Approach to Matrix Spectral Analysis. In *Proc. 9th Innovations in Theoretical Computer Science Conference*, pages 6:1–6:22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. 2, 4, 26

[26] Peter Benner, Volker Mehrmann, and Hongguo Xu. A new method for computing the stable invariant subspace of a real Hamiltonian matrix. *Journal of computational and applied mathematics*, 86(1):17–43, 1997. 26

[27] Rajendra Bhatia. Pinching, trimming, truncating, and averaging of matrices. *The American Mathematical Monthly*, 107(7):602–608, 2000. 25

[28] Rajendra Bhatia. *Perturbation bounds for matrix eigenvalues*. SIAM, 2007. 20

[29] Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *Proc. Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 902–920. SIAM, 2014. 9

[30] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014. 9

[31] Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proc. Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 236–249, 2016. 4, 10

[32] Tim Clark and Rainer Koch. *Linear Combination of Atomic Orbitals*, pages 5–22. Springer Berlin Heidelberg, 1999. 58

[33] Kenneth L Clarkson and David P Woodruff. Numerical linear algebra in the streaming model. In *Proc. Forty-First Annual ACM Symposium on Theory of Computing*, pages 205–214, 2009. 9, 64

[34] Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM*, 63(6):1–45, 2017. 64

[35] Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proc. Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 163–172, 2015. 1

[36] Charles R. Crawford. Reduction of a band-symmetric generalized eigenvalue problem. *Communications of the ACM*, 16(1):41–44, 1973. 26

[37] Philip I Davies, Nicholas J Higham, and Françoise Tisseur. Analysis of the cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 23(2):472–493, 2001. 2, 4, 26

[38] Theodorus J Dekker and Joseph F Traub. The shifted QR algorithm for Hermitian matrices. *Linear Algebra and its Applications*, 4(3):137–154, 1971. 26

[39] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007. 2, 3, 4, 9, 21, 22, 23, 34, 47, 64

[40] James Demmel, Ioana Dumitriu, Olga Holtz, and Robert Kleinberg. Fast matrix multiplication is stable. *Numerische Mathematik*, 106(2):199–224, 2007. 2, 21, 22

[41] James Demmel, Ioana Dumitriu, and Ryan Schneider. Generalized Pseudospectral Shattering and Inverse-Free Matrix Pencil Diagonalization. *arXiv preprint arXiv:2306.03700*, 2023. 2, 3, 4, 10, 27, 33

[42] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997. 2, 26

[43] James Weldon Demmel. Three methods for refining estimates of invariant subspaces. *Computing*, 38(1):43–57, 1987. 26

[44] James Weldon Demmel and Bo Kågström. Computing stable eigendecompositions of matrix pencils. *Linear Algebra and its Applications*, 88:139–186, 1987. 26

[45] Konstantinos I Diamantaras and Sun Yuan Kung. *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc., 1996. 1, 9, 62

[46] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *Proc. Thiry-Fourth Annual ACM Symposium on Theory of Computing*, pages 82–90, 2002. 1

[47] Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science*, pages 2129–2138. IEEE, 2023. 2, 21

[48] Stanley C Eisenstat and Ilse CF Ipsen. Relative perturbation results for eigenvalues and eigenvectors of diagonalisable matrices. *BIT Numerical Mathematics*, 38(3):502–509, 1998. 20

[49] Ethan N Epperly, Joel A Tropp, and Robert J Webber. XTrace: Making the Most of Every Sample in Stochastic Trace Estimation. *SIAM Journal on Matrix Analysis and Applications*, 45(1):1–23, 2024. 62

[50] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020. 1

[51] John GF Francis. The QR transformation a unitary analogue to the LR transformation—Part 1. *The Computer Journal*, 4(3):265–271, 1961. 26

[52] John GF Francis. The QR transformation—Part 2. *The Computer Journal*, 4(4):332–345, 1962. 26

[53] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *Proc. 39th Annual Symposium on Foundations of Computer Science*, page 370, 1998. 9, 64

[54] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004. 64

[55] Martin Fürer. Faster integer multiplication. In *Proc. Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 57–66, 2007. 2

[56] Giulia Galli. Linear scaling methods for electronic structure calculations and quantum molecular dynamics simulations. *Current Opinion in Solid State and Materials Science*, 1(6):864–874, 1996. 59

[57] Giulia Galli and Michele Parrinello. Large scale electronic structure calculations. *Physical Review Letters*, 69(24):3547, 1992. 59

[58] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973. 4

[59] P Giannozzi, O Andreussi, T Brumme, O Bunau, M Buongiorno Nardelli, M Calandra, R Car, C Cavazzoni, D Ceresoli, M Cococcioni, N Colonna, I Carnimeo, A Dal Corso, S de Gironcoli, P Delugas, R A DiStasio, A Ferretti, A Floris, G Fratesi, G Fugallo, R Gebauer, U Gerstmann, F Giustino, T Gorni, J Jia, M Kawamura, H-Y Ko, A Kokalj, E Küçükbenli, M Lazzeri, M Marsili, N Marzari, F Mauri, N L Nguyen, H-V Nguyen, A Otero de-la Roza, L Paulatto, S Poncé, D Rocca, R Sabatini, B Santra, M Schlipf, A P Seitsonen, A Smogunov, I Timrov, T Thonhauser, P Umari, N Vast, X Wu, and S Baroni. Advanced capabilities for

materials modelling with Quantum ESPRESSO. *Journal of Physics: Condensed Matter*, 29(46):465901, oct 2017. 8, 58

[60] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, Andrea Dal Corso, Stefano de Gironcoli, Stefano Fabris, Guido Fratesi, Ralph Gebauer, Uwe Gerstmann, Christos Gougoussis, Anton Kokalj, Michele Lazzeri, Layla Martin-Samos, Nicola Marzari, Francesco Mauri, Riccardo Mazzarello, Stefano Paolini, Alfredo Pasquarello, Lorenzo Paulatto, Carlo Sbraccia, Sandro Scandolo, Gabriele Sclauzero, Ari P Seitsonen, Alexander Smogunov, Paolo Umari, and Renata M Wentzcovitch. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, 21(39):395502, Sep 2009. 8

[61] John R Gilbert and Robert Endre Tarjan. The analysis of a nested dissection algorithm. *Numerische Mathematik*, 50(4):377–404, 1986. 4

[62] Stefan Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71(4):1085, 1999. 59

[63] Stefan Goedecker and GE Scuserza. Linear scaling electronic structure methods in chemistry and physics. *Computing in Science & Engineering*, 5(4):14–21, 2003. 59

[64] Gene H Golub and Charles F Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2013. 2, 19, 26, 64

[65] Ming Gu and Stanley C Eisenstat. A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 16(1):172–191, 1995. 10, 59

[66] Ming Gu and Stanley C Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996. 9

[67] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. 9, 64

[68] David Harvey and Joris Van Der Hoeven. Integer multiplication in time o(nlog\,n). *Annals of Mathematics*, 193(2):563–617, 2021. 2

[69] Qiu He, Bin Yu, Zhaohuai Li, and Yan Zhao. Density Functional Theory for Battery Materials. *Energy & Environmental Materials*, 2(4):264–279, 2019. 58

[70] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002. 19, 20, 61, 64

[71] Nicholas J Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):251–254, 2009. 4

[72] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933. 9, 62

[73] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 1

[74] Jürg Hutter, Marcella Iannuzzi, Florian Schiffmann, and Joost VandeVondele. cp2k: atomistic simulations of condensed matter systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(1):15–25, 2014. 8, 58

[75] Ilse CF Ipsen. Computing an eigenvector with inverse iteration. *SIAM Review*, 39(2):254–291, 1997. 2, 26

[76] Ilse CF Ipsen. Absolute and relative perturbation bounds for invariant subspaces of matrices. *Linear Algebra and its Applications*, 309(1-3):45–56, 2000. 26

[77] Zhongxiao Jia and GW Stewart. An analysis of the rayleigh–ritz method for approximating eigenspaces. *Mathematics of computation*, 70(234):637–647, 2001. 26

[78] William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.*, 26(1):189–206, 1984. 61

[79] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002. 1, 9, 62

[80] Praneeth Kacham and David P Woodruff. Faster Algorithms for Schatten-p Low Rank Approximation. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2024. 64

[81] William Kahan. Spectra of nearly Hermitian matrices. *Proc. American Mathematical Society*, 48(1):11–17, 1975. 20

[82] Daniel M Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):1–23, 2014. 62

[83] Charles S Kenney and Alan J Laub. The matrix sign function. *IEEE transactions on automatic control*, 40(8):1330–1348, 1995. 10

[84] Andrzej Kiełbasiński. A note on rounding-error analysis of Cholesky factorization. *Linear Algebra and its Applications*, 88:487–494, 1987. 4

[85] Cedric Klinkert, Áron Szabó, Christian Stieger, Davide Campi, Nicola Marzari, and Mathieu Luisier. 2-d Materials for Ultrascaled Field-Effect Transistors: One Hundred Candidates under the Ab Initio Microscope. *ACS Nano*, 14(7):8605–8615, 2020. 58

[86] Walter Kohn. Density functional/Wannier function theory for systems of very many atoms. *Chemical Physics Letters*, 208(3-4):167–172, 1993. 59

[87] Walter Kohn. Density functional and density matrix method scaling linearly with the number of atoms. *Physical Review Letters*, 76(17):3168, 1996. 59

[88] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133, 1965. 2, 58

[89] Georg Kresse and Jürgen Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical Review B*, 54(16):11169–11186, 1996. 8, 58

[90] Vera N Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics*, 1(3):637–657, 1962. 26

[91] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A Spielman. Sparsified Cholesky and Multigrid Solvers for Connection Laplacians. In *Proc. 48th Annual ACM Symposium on Theory of Computing*, pages 842–850, 2016. 4

[92] Rasmus Kyng and Sushant Sachdeva. Approximate Gaussian Elimination for Laplacians - Fast, Sparse, and Simple. In *Proc. IEEE 57th Annual Symposium on Foundations of Computer Science*, pages 573–582, 2016. 4

[93] Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson-Lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, pages 633–638. IEEE, 2017. 62

[94] Richard J Lipton, Donald J Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16(2):346–358, 1979. 4

[95] Anand Louis and Santosh S Vempala. Accelerated Newton iteration for roots of black box polynomials. In *Proc. IEEE 57th Annual Symposium on Foundations of Computer Science*, pages 732–740. IEEE, 2016. 2, 3

[96] Arnaz Malhi and Robert X Gao. PCA-based feature selection scheme for machine defect classification. *IEEE transactions on instrumentation and measurement*, 53(6):1517–1525, 2004. 1

[97] Alexander N Malyshev. Computing invariant subspaces of a regular linear pencil of matrices. *Siberian Mathematical Journal*, 30(4):559–567, 1989. 26

[98] Alexander N Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra and its Applications*, 188:489–520, 1993. 26

[99] Olvi L Mangasarian and Edward W Wild. Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):69–74, 2005. 1

[100] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47–68, 2011. 9, 64

[101] Nicola Marzari, Arash A. Mostofi, Jonathan R. Yates, Ivo Souza, and David Vanderbilt. Maximally localized Wannier functions: Theory and applications. *Reviews of Modern Physics*, 84(4):1419, 2012. 58

[102] Reinhard J Maurer, Christoph Freysoldt, Anthony M Reilly, Jan Gerit Brandenburg, Oliver T Hofmann, Torbjörn Björkman, Sébastien Lebègue, and Alexandre Tkatchenko. Advances in Density-Functional Calculations for Materials Modeling. *Annual Review of Materials Research*, 49(1):1–30, 2019. 58

[103] István Mayer. *Simple theorems, proofs, and derivations in quantum chemistry*. Springer Science & Business Media, 2003. 59

[104] Raphael Meyer, Cameron Musco, and Christopher Musco. On the unreasonable effectiveness of single vector Krylov methods for low-rank approximation. In *Proc. 2024 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 811–845. SIAM, 2024. 4

[105] Nariyuki Minami. Local fluctuation of the spectrum of a multidimensional Anderson tight binding model. *Communications in mathematical physics*, 177:709–725, 1996. 4

[106] Cameron Musco and Christopher Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. *Advances in Neural Information Processing Systems*, 28, 2015. 4, 9, 64, 65

[107] Cameron Musco, Christopher Musco, and Aaron Sidford. Stability of the lanczos method for matrix function approximation. In *Proc. Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1605–1624. SIAM, 2018. 2, 3, 24, 64

[108] Yuji Nakatsukasa. On the condition numbers of a multiple eigenvalue of a generalized eigenvalue problem. *Numerische Mathematik*, 121:531–544, 2012. 26

[109] Yuji Nakatsukasa and Nicholas J Higham. Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD. *SIAM Journal on Scientific Computing*, 35(3):A1325–A1349, 2013. 2

[110] Jelani Nelson and Huy L Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proc. IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2013. 62

[111] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001. 1

[112] Christopher C Paige. Error analysis of the lanczos algorithm for tridiagonalizing a symmetric matrix. *IMA Journal of Applied Mathematics*, 18(3):341–349, 1976. 26

[113] Christopher Conway Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices.* PhD thesis, University of London, 1971. 2, 26

[114] Victor Y Pan and Zhao Q Chen. The complexity of the matrix eigenproblem. In *Proc. 31st Annual ACM Symposium on Theory of Computing*, pages 507–516, 1999. 26

[115] Beresford N Parlett. *The Symmetric Eigenvalue Problem*. SIAM, 1998. 26

[116] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. 9, 62

[117] Andrzej Pokrzywa. Spectra of operators with fixed imaginary parts. *Proc. American Mathematical Society*, 81(3):359–364, 1981. 20

[118] Awwal Mohammed Rufai, Gholamreza Anbarjafari, and Hasan Demirel. Lossy image compression using singular value decomposition and wavelet difference reduction. *Digital signal processing*, 24:117–123, 2014. 1

[119] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011. 26

[120] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proc. 47th Annual Symposium on Foundations of Computer Science*, pages 143–152. IEEE, 2006. 9, 64

[121] Ryan Schneider. When is fast, implicit squaring of $A^{-1}B$ stable? *arXiv preprint arXiv:2310.00193*, 2023. 2, 3, 10

[122] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998. 1

[123] Arnold Schönhage and Volker Strassen. Fast multiplication of large numbers. *Computing*, 7:281–292, 1971. 2

[124] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 1

[125] Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. Tight query complexity lower bounds for PCA via finite sample deformed wigner law. In *Proc. 50th Annual ACM Symposium on Theory of Computing*, pages 1249–1259, 2018. 64

[126] Aleksandros Sobczyk and Mathieu Luisier. Approximate Euclidean lengths and distances beyond Johnson-Lindenstrauss. *Advances in Neural Information Processing Systems*, 35:19357–19369, 2022. 62

[127] José M Soler, Emilio Artacho, Julian D Gale, Alberto García, Javier Junquera, Pablo Ordejón, and Daniel Sánchez-Portal. The SIESTA method for ab initio order-N materials simulation. *Journal of Physics: Condensed Matter*, 14(11):2745, 2002. 8, 58

[128] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004. 4

[129] Nikhil Srivastava. The complexity of diagonalization. In *Proc. 2023 International Symposium on Symbolic and Algebraic Computation*, pages 1–6, 2023. 26

[130] Gilbert W Stewart. On the sensitivity of the eigenvalue problem ax=λbx. *SIAM Journal on Numerical Analysis*, 9(4):669–686, 1972. 26

[131] Gilbert W Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, 1973. 26

[132] Gilbert W Stewart. On the perturbation of pseudo-inverses, projections and linear least squares problems. *SIAM Review*, 19(4):634–662, 1977. 26

[133] Gilbert W Stewart. Pertubation bounds for the definite generalized eigenvalue problem. *Linear Algebra and its Applications*, 23:69–85, 1979. 26

[134] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969. 21

[135] Lloyd N Trefethen and Mark Embree. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005. 20, 21, 26

[136] Joost VandeVondele, Urban Borstnik, and Jurg Hutter. Linear scaling self-consistent field calculations with millions of atoms in the condensed phase. *Journal of Chemical Theory and Computation*, 8(10):3565–3573, 2012. 59

[137] James Vogel, Jianlin Xia, Stephen Cauley, and Venkataramanan Balakrishnan. Superfast divide-and-conquer method and perturbation analysis for structured eigenvalue solutions. *SIAM Journal on Scientific Computing*, 38(3):A1358–A1382, 2016. 10, 59

[138] Franz Wegner. Bounds on the density of states in disordered systems. *Zeitschrift für Physik B Condensed Matter*, 44(1):9–15, 1981. 4, 7, 47

[139] Hermann Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912. 20

[140] James Hardy Wilkinson. Global convergene of tridiagonal QR algorithm with origin shifts. *Linear Algebra and its Applications*, 1(3):409–420, 1968. 26

[141] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proc. 2024 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 3792–3835. SIAM, 2024. 2, 21

[142] Zewen Xiao and Yanfa Yan. Progress in Theoretical Study of Metal Halide Perovskite Solar Cell Materials. *Advanced Energy Materials*, 7(22):1701136, 2017. 58

[143] Weitao Yang. Direct calculation of electron density in density-functional theory. *Physical Review Letters*, 66(11):1438, 1991. 59

[144] Yunkai Zhou, Yousef Saad, Murilo L Tiago, and James R Chelikowsky. Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration. *Physical Review E*, 74(6):066704, 2006. 59

[145] Yunkai Zhou, Yousef Saad, Murilo L Tiago, and James R Chelikowsky. Self-consistent-field calculations using Chebyshev-filtered subspace iteration. *Journal of Computational Physics*, 219(1):172–184, 2006. 59

A Preliminaries

In this section the model of computation is defined and basic linear algebra principles are summarized. Further details can be found in standard textbooks such as [70, 64].

A.1 Forward and backward approximation

In numerical analysis, a "forward-error" often measures the distance between the true solution to a problem and the solution returned by a numerical algorithm. A related notion is the so-called "backward error" or "backward approximation." For more details we can refer to the standard textbook of Higham [70]. In this case, the solution returned is the exact solution of a "nearby problem," and the backward error quantifies the distance of the original problem to this nearby problem. In PCA, for example, a backward-approximation could be defined by $\widetilde{\mathbf{C}}_k = \arg\min_{\mathbf{C}^\top \mathbf{C} = \mathbf{I}_{k \times k}} \operatorname{tr}(\mathbf{C}^\top \widetilde{\mathbf{H}} \mathbf{C})$. The backward error would be given $\|\mathbf{H} - \widetilde{\mathbf{H}}\|$ for some norm. An algorithm is *backward stable* if the backward error is always well-defined and bounded. A forward error type of approximation is often harder to achieve. A common rule-of-thumb states that

$$\text{forward error} \lesssim \text{backward error} \times \text{condition number of the problem.}$$

However, this does not generally hold for eigenvalue problems. For example, Proposition 1.1 in [15] can be used to transformed a backward approximate diagonalization error to a forward error for the eigenvectors, but the bound depends on the minimum eigenvalue gap between any eigenvalue pair. As the authors point in Remark 1.2, special treatment is needed in terms of the invariant subspaces in the presence of multiple eigenvalues, which are thoroughly analyzed in this work.

A.2 Floating point model

We assume a standard floating point model of computation and borrow its axioms from [70, Chapter 2]. There is a fixed number of bits to represent floating point numbers, specifically, one bit is reserved for the sign s, p bits are used for the exponent e, and t bits are used for the significand m.

A real number $\alpha \in \mathbb{R}$ is rounded to a floating point number

$$\mathbf{fl}(\alpha) = s \times 2^{e-t} \times m.$$

The sign s is $+$ if the corresponding bit is one, and $-$ if the bit is zero. The exponent e is stored as a binary number in the so-called *biased form*, and its range is $e \in [-M, M]$, where $M = 2^{p-1}$. The significand m is an integer that satisfies $2^{t-1} \leq m \leq 2^t - 1$, where the lower bound is enforced to ensure that the system is *normalized*, i.e. the first bit of m is always 1. We can therefore write $\mathbf{fl}(\alpha)$ in a more intuitive representation

$$\mathbf{fl}(\alpha) = \pm 2^e \times \left(\frac{m_1}{2} + \frac{m_2}{2^2} + \ldots + \frac{m_t}{2^t} \right),$$

where the first bit m_1 of m is always equal to one for normalized numbers. The range of normalized numbers is therefore $[2^{-M}, 2^M(2 - 2^{-t})]$. Numbers that are smaller than 2^{-M} are called *subnormal* and they will be ignored for simplicity, since we can either add more bits in the exponent, or account for them in the failure probability when the numbers are random (the latter strategy is used for example in Lemma D.1). Similarly, numbers that are larger than $2^M(2 - 2^{-t})$ are assumed to be numerically equal to infinity, denoted by INF.

From [70, Theorem 2.2], for all real numbers α in the normalized range it holds that

$$\mathbf{fl}(\alpha) = (1 + \theta)\alpha,$$

where $\theta \in \mathbb{R}$ satisfies $|\theta| \leq 2^{-t} := \mathbf{u}$, where $\mathbf{u}$ is the *machine precision*. Clearly, $t = O(\log(1/\mathbf{u}))$, in which case we can always obtain a bound for the number of required bits of a numerical algorithm if we have an upper bound for the precision $\mathbf{u}$. We will write the same for complex numbers which are represented as a pair of normalized floating point numbers.

The floating point implementation of each arithmetic operation $\odot \in \{+, -, \times, /\}$ also satisfies

$$\mathbf{fl}(\alpha \odot \beta) = (1 + \theta)(\alpha \odot \beta), \quad |\theta| \leq \mathbf{u}. \tag{4}$$

Divisions and multiplications with 1 and 2 do not introduce errors (for the latter we simply increase/decrease the exponent). We assume that we also have an implementation of $\sqrt{\cdot}$ such that $\mathbf{fl}(\sqrt{\alpha}) = (1+\theta)\sqrt{\alpha}$ where $|\theta| \leq \mathbf{u}$. From [70, Lemma 3.1], we can bound products of errors as

$$\prod_{i=1}^{n}(1+\theta_i)^{\rho_i} = 1 + \eta_n,$$

where $\rho_i = \pm 1$ and $|\eta_n| \leq \frac{n\mathbf{u}}{1-n\mathbf{u}}$.

The above can be extended also for complex arithmetic (see [70, Lemma 3.5]), where the bound becomes $|\theta| \leq O(\mathbf{u})$, but we will ignore the constant prefactor for simplicity.

Operations on matrices can be analyzed in a similar manner. Let $\otimes$ denote the element-wise multiplication between two matrices and $\oslash$ the element-wise division. The floating point representation of a matrix $\mathbf{A}$ satisfies

$$\mathbf{fl}(\mathbf{A}) = \mathbf{A} + \mathbf{\Delta} \otimes \mathbf{A}, \quad |\mathbf{\Delta}_{i,j}| \leq \mathbf{u}.$$

It can be shown that $\|\mathbf{\Delta}\| \leq \mathbf{u}\sqrt{n}\|\mathbf{A}\|$.

For any operation $\odot \in \{+, -, \otimes, \oslash\}$ and matrices $\mathbf{A}$ and $\mathbf{B}$ it holds that

$$\mathbf{fl}(\mathbf{A} \odot \mathbf{B}) = \mathbf{A} \odot \mathbf{B} + \mathbf{\Delta} \otimes (\mathbf{A} \odot \mathbf{B}), \quad |\mathbf{\Delta}_{i,j}| \leq \mathbf{u}, \quad \|\mathbf{\Delta} \otimes (\mathbf{A} \odot \mathbf{B})\| \leq \mathbf{u}\sqrt{n}\|\mathbf{A} \odot \mathbf{B}\|. \quad (5)$$

A.3   Spectral decomposition, pseudospectrum, and eigenvalue bounds

We first recall the definition of the spectral decomposition of a diagonalizable matrix.

Definition A.1 (Spectrum and spectral decomposition). *A matrix $\mathbf{A}$ is diagonalizable if there exist invertible matrix $\mathbf{V}$ and diagonal matrix $\mathbf{\Lambda}$ such that $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$. This is called the spectral decomposition of $\mathbf{A}$. The set $\Lambda(\mathbf{A}) = \{\mathbf{\Lambda}_{i,i} | i = 1, \ldots, n\}$ is the spectrum of $\mathbf{A}$.*

The spectral theorem states that Hermitian matrices (or, more generally, normal matrices) can be always diagonalized by unitary transformations.

Theorem A.1 (Spectral theorem). *If $\mathbf{A} \in \mathbb{C}^{n \times n}$ is Hermitian, then there exists orthogonal matrix $\mathbf{Q} \in \mathbb{C}^{n \times n}$ such that $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^*$, where $\mathbf{\Lambda}$ is a diagonal matrix with real diagonal elements.*

In the sections that follow we need to bound the (forward) errors on the eigenvalues of matrices under perturbations that are introduced due to the finite precision arithmetic. Such bounds can be derived by the classic Bauer-Fike theorem the following (cf. [24, 48] for more details).

There are many bounds in the literature describing the effect of perturbations on the eigenvalues. We summarize some classic results in the following proposition and refer to Bhatia's monograph for a detailed overview [28].

Fact A.1. *Let $\mathbf{A}$ and $\mathbf{B}$ be two $n \times n$ matrices. The following bounds are known between the eigenvalues $\lambda_1(\mathbf{A}) \leq \lambda_2(\mathbf{A}) \leq \ldots \leq \lambda_n(\mathbf{A})$ of $\mathbf{A}$ and $\lambda_1(\mathbf{B}) \leq \ldots \leq \lambda_n(\mathbf{B})$ of $\mathbf{B}$:*

A	**B**	*bound*	*reference*
Hermitian	*Hermitian*	$|\lambda_i(\mathbf{A}) - \lambda_i(\mathbf{B})| \leq \|\mathbf{A} - \mathbf{B}\|$	*[139]*
Hermitian	*Non-Hermitian*	$|\lambda_i(\mathbf{A}) - \lambda_i(\mathbf{B})| \leq O(\log(n))\|\mathbf{A} - \mathbf{B}\|$	*[81, 117]*

Pseudospectral analysis is useful when the aforementioned bounds are not applicable.

Definition A.2 (Pseudospectrum). *For some $\epsilon > 0$, the ϵ-pseudospectrum of a matrix $\mathbf{M} \in \mathbb{C}^{n \times n}$ is defined as:*

$$\Lambda_\epsilon(\mathbf{M}) := \{\lambda \in \mathbb{C} : \lambda \in \Lambda(\mathbf{M} + \mathbf{E}) \text{ for some } \|\mathbf{E}\| < \epsilon\}$$
$$= \{\lambda \in \mathbb{C} : \|(\lambda\mathbf{I} - \mathbf{M})^{-1}\| > 1/\epsilon\}$$

where $\Lambda(\mathbf{M})$ is the spectrum of $\mathbf{M}$.

Recall some useful properties from the seminal book of Trefethen and Embree [135].

Proposition A.1 (Collective results from [135]). *Let $D(z, r)$ denote the open disk of radius r in the complex plane centered at $z \in \mathbb{C}$, $\mathbf{M}$ and $\mathbf{E}$ be two $n \times n$ matrices, and $\epsilon > \|\mathbf{E}\|$ be a positive real number. The following hold:*

(i) $\Lambda_{\epsilon - \|\mathbf{E}\|}(\mathbf{M}) \subseteq \Lambda_\epsilon(\mathbf{M} + \mathbf{E}) \subseteq \Lambda_{\epsilon + \|\mathbf{E}\|}(\mathbf{M})$ *([135, Thm 52.4]),*

(ii) *Any bounded connected component of $\Lambda_\epsilon(\mathbf{M})$ has a nonempty intersection with $\Lambda(\mathbf{M})$, i.e., it contains at least one eigenvalue ([135, Thm 4.3]),*

(iii) $\bigcup_i D(\lambda_i, \epsilon) \subseteq \Lambda_\epsilon(\mathbf{M}) \subseteq \bigcup_i D(\lambda_i, \epsilon \kappa_{\mathbf{V}}(\mathbf{M}))$ *([135, Thms 4.3 and 52.2]).*

A.4 Eigenvector condition number of definite pencils

Some approximation bounds throughout the paper depend on the eigenvector condition number of the generalized eigenproblem (1). For arbitrary GEPs a bound for this quantity might not always exist, but a straightforward bound exists for the Hermitian definite case. To obtain such a bound we define the eigenvector condition number of a diagonalizable matrix as follows:

$$\kappa_{\mathbf{V}}(A) := \inf_{\substack{\mathbf{V}\mathbf{D}\mathbf{V}^{-1} = \mathbf{A} \\ \mathbf{D}: \text{ diagonal}}} \|V\| \|V^{-1}\|.$$

Proposition A.2. *Let $\mathbf{H} \in \mathbb{H}^n$ and $\mathbf{S} \in \mathbb{H}^n_{++}$, and consider the definite GEP $\mathbf{HC} = \mathbf{SC\Lambda}$. Then*

$$\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H}) \leq \sqrt{\kappa(\mathbf{S})}.$$

Proof. Since $\mathbf{S}$ is Hermitian positive-definite, it can be written as $\mathbf{S} = \mathbf{LL}^*$ for some matrix $\mathbf{L}$. Then we can transform the GEP to a Hermitian eigenproblem, specifically

$$\mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-*}\mathbf{L}^*\mathbf{C} = \mathbf{L}^*\mathbf{C\Lambda}.$$

Since $\mathbf{LL}^* = \mathbf{S}$ we have that $\|\mathbf{L}\|^2 = \|\mathbf{LL}^*\| = \|\mathbf{S}\|$ and similarly $\|\mathbf{L}^{-1}\|^2 = \|\mathbf{S}^{-1}\|$, which means that $\kappa(\mathbf{L}) = \sqrt{\kappa(\mathbf{S})}$. Since $\mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-*}$ is Hermitian, it can be diagonalized by a unitary matrix, i.e. there exists $\widehat{C}$ such that $\|\mathbf{L}^*\widehat{C}\| = \|(\mathbf{L}^*\widehat{C})^{-1}\| = 1$ and $\mathbf{L}^*\widehat{C}$ diagonalizes $\mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-*}$. For $\widehat{C}$ we have that

$$\|\widehat{C}\| = \|\mathbf{L}^{-*}\mathbf{L}^*\widehat{C}\| \leq \|\mathbf{L}^{-*}\| \|\mathbf{L}^*\widehat{C}\| = \|\mathbf{L}^{-*}\|.$$

Similarly,

$$\|\widehat{C}^{-1}\| = \|\widehat{C}^{-1}\mathbf{L}^{-*}\mathbf{L}^*\| \leq \|\widehat{C}^{-1}\mathbf{L}^{-*}\| \|\mathbf{L}^*\| = \|(\mathbf{L}^*C)^{-1}\| \|\mathbf{L}^*\| = \|\mathbf{L}^*\|.$$

Undoing the transformation, we can see that $\widehat{C}$ also satisfies:

$$\mathbf{H}\widehat{C} = \mathbf{S}\widehat{C}\mathbf{\Lambda},$$

i.e. $\widehat{C}$ diagonalizes $\mathbf{S}^{-1}\mathbf{H}$ since $\widehat{C}^{-1}\mathbf{S}^{-1}\mathbf{H}\widehat{C} = \mathbf{\Lambda}$. We conclude that $\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H}) \leq \kappa(\widehat{C}) \leq \|\mathbf{L}\| \|\mathbf{L}^{-1}\| = \sqrt{\kappa(\mathbf{S})}$. $\square$

A.5 Imported floating point algorithms for fast linear algebra

Before we dive into the details, we recall the concept of "fast matrix multiplication", pioneered by Strassen [134], who showed that two square matrices can be multiplied in $O(n^\omega)$, where $\omega = \log_2 7 \approx 2.807 < 3$, in real arithmetic. Since then, the *matrix multiplication exponent* ω has been significantly reduced, the record to date being $\omega \leq 2.371552$ [47, 141]. In two seminal works [40, 39], Demmel, Dumitriu, Holtz, and Kleinberg, proved that *any* fast matrix multiplication algorithm can be executed numerically stably in a floating point machine with almost the same arithmetic complexity as in real arithmetic. They also showed that other problems in Numerical Linear Algebra can be reduced to such matrix multiplications, including inversion, LU and QR factorizations, and solving linear systems of equations. Our algorithms build on the existing results for stable fast matrix multiplication [40], inversion [39], as well as backward-approximate diagonalization and the matrix sign function [15]. We import the corresponding results from the aforementioned works in the following theorems.

Theorem A.2 (MM, stable fast matrix multiplication [40]). *For every $\eta > 0$, there exists a fast matrix multiplication algorithm* MM *which takes as input two matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n}$ *and a machine precision* $\mathbf{u} > 0$ *and returns* $\mathbf{C} \leftarrow$ MM$(\mathbf{A}, \mathbf{B})$ *such that*

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

on floating point machine with precision $\mathbf{u}$, *where* $\mu_{\mathsf{MM}}(n) = n^{c_\eta}$, *for some constant* c_η *independent of* n. *Such an algorithm is called* $\mu_{\mathsf{MM}}(n)$-*stable, and it requires* $T_{\mathsf{MM}}(n) = O(n^{\omega+\eta})$ *arithmetic operations, where* ω *is the exponent of matrix multiplication in real arithmetic.*

Theorem A.3 (INV, logarithmically-stable fast inversion [39]). *For every* $\eta > 0$, *there exists a fast inversion algorithm* INV *which takes as input an invertible matrix* $\mathbf{A} \in \mathbb{C}^{n \times n}$ *and a machine precision* $\mathbf{u} > 0$ *and returns* $\mathbf{C} \leftarrow$ INV$(\mathbf{A})$ *such that*

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

on floating point machine with precision $\mathbf{u}$, *where* $\mu_{\mathsf{INV}}(n) = O(n^{c_\eta + \log(10)})$, *for some constants* $c_{\mathsf{INV}} \leq 8$ *and* c_η *independent of* n. *Such an algorithm is called* $(\mu_{\mathsf{INV}}(n), c_{\mathsf{INV}})$-*stable, and it requires* $T_{\mathsf{INV}}(n) = O(T_{\mathsf{MM}}(n))$ *arithmetic operations, where* $T_{\mathsf{MM}}(n)$ *is the same as in Theorem A.2.*

Sign function and deflation. The next result that we import is a floating point algorithm for the matrix sign function, which, for a diagonalizable matrix $\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$ such that $\Lambda(\mathbf{A})$ does not interesect with the imaginary axis, is defined as $\text{sgn}(\mathbf{A}) = \mathbf{V}\,\text{sgn}(\boldsymbol{\Lambda})\mathbf{V}^{-1}$, where $\text{sgn}(\boldsymbol{\Lambda})$ is a diagonal matrix and each diagonal entry contains the sign of the real part of the corresponding eigenvalue. To state the main result, we first recall the definition of the *Circles of Apollonius*.

Definition A.3 (Circles of Apollonius, imported from Section 4.1 of [15]). *Let* $\alpha \in (0, 1)$. *The Circles of Apollonius* C_α *are defined as* $\mathsf{C}_\alpha = \mathsf{C}_\alpha^+ \cup \mathsf{C}_\alpha^-$,

$$\mathsf{C}_\alpha^+ = \{z \in \mathbb{C} : |m(z)| \leq \alpha\}, \quad \mathsf{C}_\alpha^+ = \{z \in \mathbb{C} : |m(z)|^{-1} \leq \alpha\},$$

where $m(z) = \frac{1-z}{1+z}$ *is the Möbius transformation taking the right half-plane to the unit disk. The disk* C_α^+ *is centered at* $\frac{1+\alpha^2}{1-\alpha^2}$ *and has radius* $\frac{2\alpha}{1-\alpha^2}$, *and* C_α^- *is its reflection with respect to the imaginary axis.*

Theorem A.4 (SGN, imported Theorem 4.9 from [15]). *There is a deterministic algorithm* SGN$(\mathbf{A}, \alpha, \eta, \epsilon)$ *which takes as input a matrix* $\mathbf{A} \in \mathbb{C}^{n \times n}$, *an accuracy parameter* $\epsilon \in (0, 1/12)$, *and parameters* $\eta \in (0, 1)$, $0 < 1 - \alpha < 1/100$, *such that it is guaranteed that* $\Lambda_\eta(\mathbf{A}) \subset \mathsf{C}_\alpha$. *The algorithm returns a matrix* $\widetilde{\mathbf{S}}$ *such that*

$$\|\widetilde{\mathbf{S}} - \text{sgn}(\mathbf{A})\| \leq \epsilon,$$

as long as the machine precision satisfies

$$\mathbf{u} \leq \mathbf{u}_{\mathsf{SGN}} := \frac{\alpha^{2^{N+1}(c_{\mathsf{INV}} \log(n) + 3)}}{\mu_{\mathsf{INV}}(n)\sqrt{n}N},$$

corresponding to at most

$$O\left(\log(n) \log^3(\tfrac{1}{1-\alpha}) \left(\log(\tfrac{1}{\epsilon}) + \log(\tfrac{1}{\eta})\right)\right)$$

bits of precision. Here

$$N = \lceil \log(\tfrac{1}{1-\alpha}) + 3\log(\log(\tfrac{1}{1-\alpha})) + \log(\log(\tfrac{1}{\epsilon\eta})) + 7.79 \rceil$$

denotes the number of iterations that the algorithm executes. The arithmetic complexity is

$$O(NT_{\mathsf{MM}}(n))).$$

In PCA, we will need to be able to compute a basis for the column space of an approximate spectral projector. For this, we recall the following algorithm.

22

Theorem A.5 (DEFLATE, imported Theorem 5.3 of [15])**.** *There exists a randomized algorithm* $\widetilde{\mathbf{C}}_k \leftarrow \mathsf{DEFLATE}(\widetilde{\mathbf{\Pi}}_k, k, \beta, \epsilon)$*, which takes as input a matrix* $\widetilde{\mathbf{\Pi}}_k$*, a rank parameter* $k \in [n]$*, a parameter* β *such that* $\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\| \leq \beta$*, for some projector matrix* $\mathbf{\Pi}_k$ *of rank-k, and a desired accuracy* ϵ*, and returns a (complex) matrix* $\widetilde{\mathbf{C}}_k$ *such that, there exists a matrix* $\mathbf{C}_k$ *with* k *columns that forms an orthonormal basis for* $\mathrm{im}(\mathbf{\Pi}_k)$*, and*

$$\|\widetilde{\mathbf{C}}_k - \mathbf{C}_k\| \leq \epsilon.$$

The algorithm succeeds with probability at least $1 - \frac{(20n)^3 \sqrt{\beta}}{\epsilon^2}$*. The number of arithmetic operations is at most* $O(T_{\mathsf{MM}}(n))$*, and it requires* $O(\log(n) + \log(\frac{1}{\epsilon}))$ *bits of precision. Internally, the algorithm generates* $O(n^2)$ *random numbers to form a* $n \times n$ *complex Ginibre matrix.*[1]

Computing an orthonormal basis and the spectral norm. Lastly, we recall the following two results. The first one is for the QR factorization from [39], which we adapt suitably for our analysis, specifically, to compute an orthonormal basis for the column space of a rectangular matrix.

Theorem A.6 (Basis computation, follows from Section 4 in [39])**.** *Let* $\mathbf{A} \in \mathbb{R}^{m \times n}, m \geq n$*. There exists an algorithm* $\widetilde{\mathbf{Q}}, \widetilde{\mathbf{R}} = \mathsf{QR}(\mathbf{A})$*, which returns a matrix* $\widetilde{\mathbf{Q}} \in \mathbb{R}^{m \times n}$ *and an upper triangular matrix* $\widetilde{\mathbf{R}} \in \mathbb{R}^{n \times n}$ *in* $O(mn^{\omega - 1})$ *floating point operations using* $O(\log(\frac{n\kappa(\mathbf{A})}{\epsilon_{\mathsf{QR}}}))$ *bits, where* $\epsilon_{\mathsf{QR}} \in (0, 1)$ *is a given accuracy. The matrix* $\widetilde{\mathbf{Q}}$ *satisfies*

$$\|\mathbf{Q} - \widetilde{\mathbf{Q}}\mathbf{\Phi}\| \leq \epsilon_{\mathsf{QR}},$$

for some orthogonal matrix $\mathbf{\Phi} \in \mathbb{R}^{n \times n}$*, where* $\mathbf{Q}$ *has orthonormal columns and* $\mathbf{A} = \mathbf{QR}$ *is the true economy-QR of* $\mathbf{A}$*.*

Proof. We first scale $\mathbf{A}' \leftarrow \mathbf{A}/M$, where M is the smallest power of two that is larger than $n\|\mathbf{A}\|_{\max}$. This ensures that $\Omega(1/n) \leq \|\mathbf{A}'\| \leq 1$. Since we scale by a power of 2, then there are no floating point errors and the orthonormal basis from the QR factorization remains the same (only the upper triangular factor is scaled).

We now use the corresponding algorithm of [39] on $\mathbf{A}'$. It returns three matrices: an upper triangular matrix $\widetilde{\mathbf{R}} \in \mathbb{R}^{n \times n}$, a matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ with $\|\mathbf{W}\| \in O(n)$, and a matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$ with $\|\mathbf{Y}\| \in O(n)$. The matrix $\mathbf{\Psi} = \mathbf{I} - \mathbf{W}\mathbf{Y}^{\top} + \mathbf{E}_{\mathbf{\Psi}}$ exactly satisfies $\mathbf{\Psi}\mathbf{\Psi}^{\top} = \mathbf{I}$, and the error matrix satisfies $\|\mathbf{E}_{\mathbf{\Psi}}\| \in O(\mathrm{poly}(n)\mathbf{u})$. Moreover, $\mathbf{\Psi} \begin{pmatrix} \widetilde{\mathbf{R}} \\ 0_{m-n \times n} \end{pmatrix} = \widehat{\mathbf{A}}$, where $\widehat{\mathbf{A}} = \mathbf{A}' + \mathbf{E}_{\mathbf{A}}$ and $\|\mathbf{E}_{\mathbf{A}}\| \leq O(\mathrm{poly}(n)\mathbf{u})\|\mathbf{A}'\| \leq O(\mathrm{poly}(n)\mathbf{u})$. Note that $\|\widetilde{\mathbf{R}}\| \leq \|\widehat{\mathbf{A}}\| \leq \|\mathbf{A}'\| + \|\mathbf{E}_{\mathbf{A}}\| \leq 1 + O(\mathrm{poly}(n)\mathbf{u})$. The total cost of the algorithm is $O(mn^{\omega - 1})$ floating point operations (ignoring the negligible term $\eta > 0$ in the exponent).

Using $\mathbf{W}$ and $\mathbf{Y}$, we can construct an approximate basis $\widetilde{\mathbf{Q}}$ as follows. Let $\mathbf{Y}_n \in \mathbb{R}^{n \times n}$ contain the first n columns of $\mathbf{Y}$. We can compute the matrix $\widetilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{I}_n \\ 0_{m-n \times n} \end{pmatrix} - \mathsf{MM}(\mathbf{W}, \mathbf{Y}_n)$ in $O(mn^{\omega - 1})$ floating point operations by performing the multiplication in blocks of size $n \times n$. We can write $\widetilde{\mathbf{Q}} = \widehat{\mathbf{Q}} + \mathbf{E}_{\widehat{\mathbf{Q}}}$ where $\widehat{\mathbf{Q}}$ contains the first n columns of $\mathbf{\Psi}$ and $\|\mathbf{E}_{\widehat{\mathbf{Q}}}\| \in O(\mathrm{poly}(n)\mathbf{u})$.

Note that $\widetilde{\mathbf{R}}$ satisfies $\widehat{\mathbf{Q}}\widetilde{\mathbf{R}} = \mathbf{A}' + \mathbf{E}_{\mathbf{A}}$. Since $\widehat{\mathbf{Q}}$ has orthonormal columns, the singular values of $\widetilde{\mathbf{R}}$ satisfy $\sigma_i(\widetilde{\mathbf{R}}) = \sigma_i(\widehat{\mathbf{Q}}\widetilde{\mathbf{R}}) = \sigma_i(\mathbf{A}' + \mathbf{E}_{\mathbf{A}}) \in \sigma_i(\mathbf{A}') \pm \|\mathbf{E}_{\mathbf{A}}\|$, where the last comes from the stability of singular values which is a consequence of Weyl's inequality in Fact A.1. Now, if $\|\mathbf{E}_{\mathbf{A}}\| \leq \frac{1}{4}\sigma_{\min}(\mathbf{A}')$, which is achieved by setting $\mathbf{u} \leq c\frac{1}{\mathrm{poly}(n)\kappa(\mathbf{A})}$ for some constant c, then $\widetilde{\mathbf{R}}$ is full rank, and it holds that $\|\widetilde{\mathbf{R}}^{-1}\| = 1/\sigma_{\min}(\widetilde{\mathbf{R}}) \leq \frac{4}{3\sigma_{\min}(\mathbf{A}')} \leq O(n)\kappa(\mathbf{A})$.

Since $\widetilde{\mathbf{R}}$ remains full rank, we have that $\widehat{\mathbf{Q}}\widetilde{\mathbf{R}} = \widehat{\mathbf{A}} \Rightarrow \widehat{\mathbf{Q}} = \mathbf{A}'\widetilde{\mathbf{R}}^{-1} + \mathbf{E}_{\mathbf{A}}\widetilde{\mathbf{R}}^{-1}$. If we write the true economy-QR of $\mathbf{A}'$ as $\mathbf{A}' = \mathbf{QR}$, then we can see that $\|\mathbf{R}\widetilde{\mathbf{R}}^{-1}\| = \|\mathbf{QR}\widetilde{\mathbf{R}}^{-1}\| = \|\mathbf{A}\widetilde{\mathbf{R}}^{-1}\| = $

---

[1] We highlight that our Gaussian sampling oracle in Definition D.1 is slightly different than the one described in [15], but it should be straightforward to adapt the analysis using either of the two definitions.

$\|\widehat{Q} - \mathbf{E_A}\widetilde{\mathbf{R}}^{-1}\|$. This implies that $\|\mathbf{R}\widetilde{\mathbf{R}}^{-1}\| \in [1 - \|\mathbf{E_A}\widetilde{\mathbf{R}}^{-1}\|, 1 + \|\mathbf{E_A}\widetilde{\mathbf{R}}^{-1}\|]$. With similar arguments we obtain

$$\|\widetilde{\mathbf{R}}\mathbf{R}^{-1}\| \in [1 - \|\mathbf{E_A}\mathbf{R}^{-1}\|, 1 + \|\mathbf{E_A}\mathbf{R}^{-1}\|].$$

We previously argued that $\|\mathbf{R}\|, \|\widetilde{\mathbf{R}}\| \leq O(1)$, and $\|\mathbf{R}^{-1}\|, \|\widetilde{\mathbf{R}}^{-1}\| \leq O(n)\kappa(\mathbf{A})$. Thus, if we further enforce $\|\mathbf{E_A}\| \leq \epsilon_1 c_1 \frac{1}{\text{poly}(n)\kappa(\mathbf{A})}$ for some constant c_1, then $\max\{\|\mathbf{E_A}\widetilde{\mathbf{R}}^{-1}\|, \|\mathbf{E_A}\mathbf{R}^{-1}\|\} \leq \epsilon_1$, for some $\epsilon_1 \in (0, 1/4)$, which means that all the singular values of $\mathbf{R}\widetilde{\mathbf{R}}^{-1}$ are inside the interval $[1 - 2\epsilon_1, 1 + 2\epsilon_1]$. In other words, $\mathbf{R}\widetilde{\mathbf{R}}^{-1}$ is approximately orthogonal. Indeed, we can write $\mathbf{R}\widetilde{\mathbf{R}}^{-1} = \mathbf{\Phi} + \mathbf{E_\Phi}$ where $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{I}$ and all the singular values of $\mathbf{E_\Phi}$ are inside $[-2\epsilon_1, 2\epsilon_1]$. In particular, $\mathbf{\Phi} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}, \mathbf{V}$ come from the SVD of $\mathbf{R}\widetilde{\mathbf{R}}^{-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.

We can now go back to $\widetilde{\mathbf{Q}}$. From the above, $\widehat{Q}$ can be written as

$$\widehat{Q} = \widehat{A}\widetilde{\mathbf{R}}^{-1} + \mathbf{E_A}\widetilde{\mathbf{R}}^{-1} = \mathbf{Q}\mathbf{\Phi} + \mathbf{Q}\mathbf{E_\Phi} + \mathbf{E_A}\widetilde{\mathbf{R}}^{-1},$$

which ultimately gives

$$\widetilde{\mathbf{Q}} = \widehat{Q} + \mathbf{E}_{\widehat{Q}} = \mathbf{Q}\mathbf{\Phi} + \mathbf{Q}\mathbf{E_\Phi} + \mathbf{E_A}\widetilde{\mathbf{R}}^{-1} + \mathbf{E}_{\widehat{Q}}.$$

This means that $\widetilde{\mathbf{Q}}$ is just a rotation of the true $\mathbf{Q}$ plus some additive error terms that we can control with $\mathbf{u}$. In particular,

$$\begin{aligned}\|\widetilde{\mathbf{Q}} - \mathbf{Q}\mathbf{\Phi}\| &\leq \|\mathbf{Q}\mathbf{E_\Phi} + \mathbf{E_A}\widetilde{\mathbf{R}}^{-1} + \mathbf{E}_{\widehat{Q}}\| \\ &\leq \epsilon_1 + O(\text{poly}(n)\mathbf{u})\|\widetilde{\mathbf{R}}^{-1}\| + O(\text{poly}(n)\mathbf{u}) \\ &\leq O(\epsilon_1),\end{aligned}$$

where in the last we applied the required bound for $\mathbf{u} \leq \epsilon_1 c_1 \frac{1}{\text{poly}(n)\kappa(\mathbf{A})}$.

Combining everything, we conclude that if the machine precision satisfies

$$\mathbf{u} \leq \epsilon_{\mathsf{QR}} \frac{1}{\text{poly}(n)\kappa(\mathbf{A})},$$

then we can compute a matrix $\widetilde{\mathbf{Q}}$ in $O(mn^{\omega-1})$ floating point operations such that there exists orthogonal matrix $\mathbf{\Phi}$ satisfying

$$\|\widetilde{\mathbf{Q}} - \mathbf{Q}\mathbf{\Phi}\| \leq \epsilon_{\mathsf{QR}},$$

using $O(\log(n\kappa(\mathbf{A})/\epsilon_{\mathsf{QR}}))$ bits of precision. $\qquad\square$

The second result is for the computation of the spectral norm of the matrix from [107] using the Lanczos algorithm.

Theorem A.7 (Imported variant of Theorem 18 from the full version of [107])**.** *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\delta \in (0, 1/2)$ a failure probability parameter. We can compute a vector $\mathbf{y}$ such that, with probability at least $1 - \delta$,*

$$\frac{9}{10}\|\mathbf{A}\| \leq \frac{\|\mathbf{A}\mathbf{y}\|}{\|\mathbf{y}\|} \leq \|\mathbf{A}\|,$$

in $O\left(mn\log(n)\log(1/\delta)\right)$ floating point operations using $O(\log(n))$ bits of precision. Internally the algorithm generates $O(n\log(1/\delta))$ random bits.

We can obtain the following corollary.

Corollary A.1 (Spectral norm)**.** *Let $\mathbf{B} \in \mathbb{C}^{m \times n}$. We can compute a value $\widetilde{\Sigma} \in \Theta(\|\mathbf{B}\|)$ with probability at least $1 - 2\delta$ in $O\left(mn\log(n)\log(1/\delta)\right)$ floating point operations using $O(\log(n))$ bits of precision.*

Proof. Let $\mathbf{R}$ be the real part of $\mathbf{B}$ and $\mathbf{Z}$ be the imaginary part. We can approximate $R \in [\frac{9}{10}\|\mathbf{R}\|, \|\mathbf{R}\|]$ and $Z \in [\frac{9}{10}\|\mathbf{Z}\|, \|\mathbf{Z}\|]$ using the algorithm of Theorem A.7. We then set $\widetilde{\Sigma} = R + Z \in [0.9(\|\mathbf{R}\| + \|\mathbf{Z}\|), \|\mathbf{R}\| + \|\mathbf{Z}\|]$. Using the triangle inequality we have that

$$0.9\|\mathbf{B}\| \leq 0.9(\|\mathbf{R}\| + \|\mathbf{Z}\|) \leq \widetilde{\Sigma} \leq \|\mathbf{R}\| + \|\mathbf{Z}\| \leq 2\|\mathbf{B}\|,$$

or, in other words, $\widetilde{\Sigma} \in [0.9\|\mathbf{B}\|, 2\|\mathbf{B}\|]$. $\qquad\square$

A.6 Symmetrization

There are certain cases where it is crucial to ensure that the result of a floating point multiplication or inversion remains Hermitian. The following proposition states that we can always "symmetrize" a floating point matrix with small additional errors.

Proposition A.3 (HERM). *Let $\mathbf{C}$ be a Hermitian matrix and $\mathbf{C}' = \mathbf{C} + \mathbf{E}$ for some (non-Hermitian) matrix $\mathbf{E}$, such that $\|\mathbf{C} - \mathbf{C}'\| = \|\mathbf{E}\|$. Consider the matrix $\mathsf{HERM}(\mathbf{C}')$, where $\mathsf{HERM}(\mathbf{A})$ is an algorithm that replaces the strictly lower triangular part of $\mathbf{A}$ with the strictly upper triangular part of $\mathbf{A}$. Then $\mathsf{HERM}(\mathbf{C}')$ is Hermitian and it holds that:*

$$\|\mathbf{C} - \mathsf{HERM}(\mathbf{C}')\| \leq c_{\mathsf{HERM}} \log(n) \cdot \|\mathbf{E}\|,$$

for some constant c_{HERM}.

Proof. For any matrix $\mathbf{A}$, the upper triangular part of $\mathbf{A}$, denoted as $\Delta_U(\mathbf{A})$, satisfies the following inequality:

$$\|\Delta_U(\mathbf{A})\| \leq c \log(n) \cdot \|\mathbf{A}\|,$$

for some constant c. The same holds for the lower triangular part $\Delta_L(\mathbf{A})$. For the diagonal part $\mathrm{diag}(\mathbf{A}))$ it holds that $\|\mathrm{diag}(\mathbf{A})\| \leq \|\mathbf{A}\|$ (see [27] for proofs). Let $\mathbf{C}'' = \mathsf{HERM}(\mathbf{C}')$. By the definition of $\mathsf{HERM}(\cdot)$, the matrix $\mathbf{C}''$ can be written as $\mathbf{C}'' = \Delta_U(\mathbf{C}') + \Delta_U(\mathbf{C}')^* - \mathrm{diag}(\mathbf{C}') = \mathbf{C} + (\Delta_U(\mathbf{E}) + \Delta_U(\mathbf{E})^* - \mathrm{diag}(\mathbf{E}))$. Therefore

$$\|\mathbf{C}'' - \mathbf{C}\| = \|\Delta_U(\mathbf{E}) + \Delta_U(\mathbf{E})^* - \mathrm{diag}(\mathbf{E})\| \leq 2c \log(n)\|\mathbf{E}\| + \|\mathbf{E}\| \leq c_{\mathsf{HERM}} \log(n)\|\mathbf{E}\|,$$

for some constant c_{HERM}. $\qquad\square$

We can directly use this to derive bounds for matrix multiplication where the result is a Hermitian matrix and for Hermitian matrix inversion.

Corollary A.2 (Symmetrized matrix multiplication). *Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices and $\mathbf{C} = \mathbf{AB}$ be a Hermitian matrix. If MM is a μ_{MM}-stable matrix multiplication algorithm then the matrix*

$$\mathbf{C}' = \mathsf{HERM}(\mathsf{MM}(\mathbf{A}, \mathbf{B}))$$

is Hermitian and it satisfies

$$\|\mathbf{C}' - \mathbf{C}\| \leq c_{\mathsf{HERM}} \log(n) \cdot \mathbf{u}\|\mathbf{A}\|\|\mathbf{B}\| \cdot \mu_{\mathsf{MM}}(n).$$

Proof. Straightforward combination of Theorem A.2 and Proposition A.3. $\qquad\square$

Corollary A.3 (Symmetrized inversion). *Let $\mathbf{A}$ be an invertible Hermitian matrix. If INV is a $(\mu_{\mathsf{INV}}, c_{\mathsf{INV}})$-stable inversion algorithm then the matrix*

$$\mathbf{C} = \mathsf{HERM}(\mathsf{INV}(\mathbf{A}))$$

is Hermitian and it satisfies

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

In addition, if $\mathbf{u}$ satisfies

$$\mathbf{u} \leq \epsilon \frac{1}{c_{\mathsf{HERM}} \log(n) \cdot \mu_{\mathsf{INV}}(n) \cdot \kappa(\mathbf{A})^{c_{\mathsf{INV}} \log(n)+1}}$$

for some $\epsilon \in (0, 1/2)$, then both of the following hold:

$$\|\mathbf{C} - \mathbf{A}^{-1}\| \leq \epsilon\|\mathbf{A}^{-1}\|,$$
$$\frac{1}{2}\kappa(\mathbf{A}) \leq \kappa(\mathbf{C}) \leq 2\kappa(\mathbf{A}).$$

Proof. The first part is straightforward combination of Theorem A.3 and Proposition A.3. For the second part, we directly bound

$$(1-\epsilon)\|\mathbf{A}^{-1}\| \leq (1-\epsilon\tfrac{1}{\kappa(\mathbf{A})})\|\mathbf{A}^{-1}\| \leq \|\mathbf{C}\| \leq (1+\epsilon\tfrac{1}{\kappa(\mathbf{A})})\|\mathbf{A}^{-1}\| \leq (1+\epsilon)\|\mathbf{A}^{-1}\|$$
$$\Rightarrow \tfrac{1}{2}\|\mathbf{A}^{-1}\| \leq \|\mathbf{C}\| \leq 2\|\mathbf{A}^{-1}\|.$$

Note that $\|\mathbf{C} - \mathbf{A}^{-1}\| \leq \epsilon\tfrac{1}{\kappa(\mathbf{A})}\|\mathbf{A}^{-1}\| = \epsilon\tfrac{1}{\|\mathbf{A}\|} = \epsilon\sigma_{\min}(\mathbf{A}^{-1})$. Since $\mathbf{C}$ and $\mathbf{A}^{-1}$ are both Hermitian then from Weyl's inequality

$$(1-\epsilon)\sigma_{\min}(\mathbf{A}^{-1}) \leq \sigma_{\min}(\mathbf{C}) \leq (1+\epsilon)\sigma_{\min}(\mathbf{A}^{-1})$$
$$\Rightarrow \tfrac{1}{2}\sigma_{\min}(\mathbf{A}^{-1}) \leq \sigma_{\min}(\mathbf{C}) \leq 2\sigma_{\min}(\mathbf{A}^{-1})$$

where we conclude that

$$\tfrac{1}{4}\kappa(\mathbf{A}) \leq \kappa(\mathbf{C}) \leq 4\kappa(\mathbf{A}).$$

$\square$

Remark A.1. *Throughout this paper, to simplify our bounds, we assume that*

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

Moreover, we define a global upper bound μ such that

$$\mu_{\mathsf{MM}}(n), c_{\mathsf{HERM}}\log(n)\mu_{\mathsf{MM}}(n), \mu_{\mathsf{INV}}(n) \leq \mu(n) \leq O(n^{c'_\eta}),$$

where, from Theorem A.2, c'_η is a constant that does not depend on n.

A.7 Existing algorithms for invariant subspaces and the (generalized) eigenproblem

We give a brief overview of some results related to the computation of spectral projectors, eigenvectors, and invariant subspaces for the Hermitian definite generalized eigenproblem. Many more details can be found in standard textbooks and references therein [42, 119, 64, 135, 115], as well as the recent overview of [129] for the complexity of the eigenproblem. For further reading we can highlight some influential works from the enormous bibliography on eigenvalue algorithms and perturbation bounds [44, 43, 8, 130, 131, 132, 133, 75, 108, 9, 26, 76, 77, 37, 2, 113, 112].

The prominent classic method to solve the Hermitian (generalized or regular) eigenproblem is to use numerically stable Householder transformations to reduce a Hermitian matrix to tridiagonal form, and then apply a tridiagonal shifted QR algorithm to diagonalize the tridiagonal matrix, which gives both the eigenvalues and the eigenvectors. The paramount QR algorithm was originally proposed in [51, 52, 90], its convergence for the symmetric case was analyzed in [140, 38] in exact arithmetic, and the non-symmetric was recently analyzed in floating point in [18, 16, 17]. Other classical algorithms for computing invariant subspaces include [97, 98, 10, 9].

With the aforementioned procedure, the floating point complexity is about $\widetilde{O}(n^3)$ floating point operations, potentially up to $\mathrm{polylog}(1/\epsilon, 1/\mathrm{gap}_k)$ factors. It is known that for Hermitian definite pencils $(\mathbf{H}, \mathbf{S})$ one can extend this procedure such that, simultaneously, $\mathbf{H}$ will become (Hermitian) tridiagonal and $\mathbf{S}$ the identity [36]. The Hermitian tridiagonal matrix can be further reduced with similarity transformations to a symmetric tridiagonal matrix, in which case one can apply again tridiagonal symmetric QR. Using standard perturbation bounds as the ones used in this work one can potentially obtain provable forward errors invariant subspaces.

While there exist works that overcome the $\widetilde{O}(n^3)$ worst-case complexity barrier for some computations related to the eigenproblem, none of them provide end-to-end forward errors for individual invariant subspaces. [114] showed that the eigenvalues of a matrix can be computed in $O(n^\omega)$ arithmetic operations, however, it becomes $O(n^{\omega+1})$ boolean operations in rational arithmetic (up to some other omitted factors), and therefore slower than standard $\widetilde{O}(n^3)$ floating point eigensolvers. [25] proposed a quantum-inspired method for approximate diagonalization in a backward-error sense, with $O(n^{\omega+1})$ bit complexity. [15] proved that backward-approximate diagonalization can be solved in $\widetilde{O}(n^\omega)$ bit complexity in floating point by using smoothed analysis, improving both the $\widetilde{O}(n^3)$ classic algorithms for the Hermitian case and the $O(\mathrm{poly}(n/\epsilon))$ algorithm of [7] for

the non-Hermitian case. This was extended for the generalized eigenproblem case in [41]. As already mentioned in the introduction, to obtain forward errors for eigenvectors from the backward-approximate solution, the corresponding bounds that are reported in [15, 41] require the existence of a minimum eigenvalue gap in the original matrix, while the analysis for invariant subspaces was left as an open problem. Our work completes this analysis, and takes a step even further, by showing that (at least based on existing results), explicit diagonalization is redundant for approximating individual invariant subspaces.

B Proof of Proposition 2.1

<div style="border:1px solid black; padding:1em;">

PURIFY

Input: Hermitian definite pencil $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}^n_{++}$, approximate condition number $\widetilde{\kappa}$ of $\mathbf{S}$, approximate values $\widetilde{\mu}_k$, $\widetilde{\mathrm{gap}}_k$, accuracy $\epsilon \in (0, 1/12)$.

Requires: $\|\mathbf{H}\| \leq 1$, $\|\mathbf{S}^{-1}\| \leq 1$, $\widetilde{\kappa} \in \Theta(\kappa(\mathbf{S}))$, $\widetilde{\mu}_k \in \mu_k \pm \frac{1}{8}\mathrm{gap}_k$, and $\widetilde{\mathrm{gap}}_k \in (1 \pm \frac{1}{8})\mathrm{gap}_k$, where $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$ and $\mathrm{gap}_k = \lambda_k - \lambda_{k+1}$ for some $k \in [n-1]$.

Algorithm: $\widetilde{\mathbf{P}} \leftarrow \mathrm{PURIFY}(\mathbf{H}, \mathbf{S}, \widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k, \widetilde{\kappa}, \epsilon)$.

1: $\mathbf{S}_{\mathsf{INV}} \leftarrow \mathsf{INV}(\mathbf{S})$. $\triangleright \mathbf{S}_{INV} = \mathbf{S}^{-1} + \mathbf{E}_1^{\mathsf{INV}}$

2: $\widetilde{\mathbf{H}} \leftarrow \mathsf{MM}(\mathbf{S}_{\mathsf{INV}}, \mathbf{H})$. $\triangleright \widetilde{\mathbf{H}} = \mathbf{S}_{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{MM}}$

3: $\widetilde{\mathbf{M}} \leftarrow \widetilde{\mu}_k - \widetilde{\mathbf{H}} + \mathbf{E}_3^{(-)}$.

4: $\widetilde{\mathbf{C}} \leftarrow \mathsf{SGN}\left(\widetilde{\mathbf{M}}, \frac{\epsilon}{16 \cdot 199}, \frac{\widetilde{\mathrm{gap}}_k}{32}\right)$. $\triangleright \widetilde{\mathbf{C}} = \mathrm{sgn}(\widetilde{\mathbf{M}}) + \mathbf{E}_4^{\mathsf{SGN}}$

5: $\widetilde{\mathbf{\Pi}} \leftarrow \frac{1}{2}\left(1 + \widetilde{\mathbf{C}}\right) + \mathbf{E}_5^{(+)}$

6: **return** $\widetilde{\mathbf{\Pi}}$.

Output: Approximate spectral projector $\widetilde{\mathbf{\Pi}}$.

Ensures: $\|\widetilde{\mathbf{\Pi}} - \mathbf{\Pi}\| \leq \epsilon$.

</div>

Algorithm 3: PURIFY.

For the analysis of Algorithm 3 we first recall that, in exact arithmetic, small perturbations do not have a severe influence in the sign function of a matrix.

Lemma B.1. *Let* $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}^n_{++}$, *where* μ *is a scalar, and* $\mathbf{M} = \mu - \mathbf{S}^{-1}\mathbf{H}$ *and* $\widetilde{\mathbf{M}} = \mu - \mathbf{S}^{-1}\mathbf{H} + \mathbf{E}$. *If* $\|\mathbf{E}\| \leq \epsilon \frac{|\lambda_{\min}(\mathbf{M})|^2 \pi}{128 \kappa_{\mathbf{V}}(\mathbf{M})}$ *for some* $\epsilon \in (0,1)$, *where* $|\lambda_{\min}(\mathbf{M})|$ *is the smallest eigenvalue of* $\mathbf{M}$ *in absolute value, then* $\|\mathrm{sgn}(\mathbf{M}) - \mathrm{sgn}(\widetilde{\mathbf{M}})\| \leq \epsilon$.

Proof. The proof uses standard techniques from holomorphic functional calculus and the properties of the pseudospectrum, but it is stated for completeness. A similar proof, for example, can be found in [9] for general matrices $\mathbf{A}$ (i.e. in that proof $\mathbf{A}$ can have complex eigenvalues).

Recall that for a matrix $\mathbf{A}$ that has no eigenvalues on the imaginary axis, it holds that
$$\mathrm{sgn}(\mathbf{A}) = \mathbf{P}_+(\mathbf{A}) - \mathbf{P}_-(\mathbf{A}),$$
where $\mathbf{P}_-(\mathbf{A})$ is the spectral projector on the subspace spanned by the eigenvectors corresponding to eigenvalues with negative real part, and $\mathbf{P}_+(\mathbf{A})$ is the corresponding spectral projector to the positive halfplane. Then
$$\|\mathrm{sgn}(\mathbf{M}) - \mathrm{sgn}(\widetilde{\mathbf{M}})\| = \|\mathbf{P}_+(\mathbf{M}) + \mathbf{P}_-(\mathbf{M}) - \mathbf{P}_+(\mathbf{M}+\mathbf{E}) - \mathbf{P}_-(\mathbf{M}+\mathbf{E})\|$$
$$\leq \|\mathbf{P}_+(\mathbf{M}) - \mathbf{P}_+(\mathbf{M}+\mathbf{E})\| + \|\mathbf{P}_-(\mathbf{M}) - \mathbf{P}_-(\mathbf{M}+\mathbf{E})\|.$$

$\mathbf{S}^{-1}\mathbf{H}$ has real eigenvalues because it is similar to a Hermitian matrix, i.e. $\Lambda(\mathbf{S}^{-1}\mathbf{H}) = \Lambda(\mathbf{L}^*\mathbf{H}\mathbf{L})$ where $\mathbf{L}$ is the lower triangular Cholesky factor of $\mathbf{S}^{-1}$. Thus $\mathbf{M}$ has also real eigenvalues. From Proposition A.1 (i) and (iii)
$$\Lambda_{\|\mathbf{E}\|}(\widetilde{\mathbf{M}}) \subseteq \Lambda_{2\|\mathbf{E}\|}(\mathbf{M}) \subseteq \bigcup_i D\left(\lambda_i(\mathbf{M}), 2\|\mathbf{E}\|\kappa_{\mathbf{V}}(\mathbf{M})\right).$$

If $\|\mathbf{E}\| \leq \epsilon_1 \frac{|\lambda_{\min}(\mathbf{M})|}{4\kappa_{\mathbf{V}}(\mathbf{M})}$ for some $\epsilon_1 \in (0,1)$ then $|\lambda_{\min}(\widetilde{\mathbf{M}})| \geq |\lambda_{\min}(\mathbf{M})|/2$ and $|\lambda_{\max}(\widetilde{\mathbf{M}})| \leq |\lambda_{\max}(\mathbf{M})| + |\lambda_{\min}(\mathbf{M})|/2$.

Let R be a rectangle whose bottom-left corner is located at $-i|\lambda_{\min}(\mathbf{M})|$ and its top-right corner at $2 + |\lambda_{\min}(\mathbf{M})| + i|\lambda_{\min}(\mathbf{M})|$. Any point on the boundary of R has a distance of at least $|\lambda_{\min}(\mathbf{M})|$ from $\Lambda(\mathbf{M})$ and at least at least $|\lambda_{\min}(\mathbf{M})|/2$ from $\Lambda(\widetilde{\mathbf{M}})$. From holomorphic functional calculus we have that

$$\mathbf{P}_+(\mathbf{A}) = \oint_{\partial R} (z - \mathbf{M})^{-1} dz,$$

where in this case ∂R denotes a positively oriented rectifiable curve over the boundary of the rectangle. Using the resolvent identity $(z - \mathbf{M})^{-1} - (z - \mathbf{M}')^{-1} = (z - \mathbf{M})^{-1}(\mathbf{M} - \mathbf{M}')(z - \mathbf{M}')^{-1}$, for $\mathbf{P}_+$ we have

$$
\begin{aligned}
\|\mathbf{P}_+(\mathbf{M}) - \mathbf{P}_+(\mathbf{M} + \mathbf{E})\| &= \left\| \frac{1}{2\pi i} \oint_{\partial R} (z - \mathbf{M})^{-1} dz - \frac{1}{2\pi i} \oint_{\partial R} (z - \mathbf{M} + \mathbf{E})^{-1} dz \right\| \\
&= \left\| \frac{1}{2\pi i} \oint_{\partial R} (z - \mathbf{M})^{-1} - (z - \mathbf{M} + \mathbf{E})^{-1} dz \right\| \\
&= \left\| \frac{1}{2\pi i} \oint_{\partial R} (z - \mathbf{M})^{-1}(z - \mathbf{M} - (z - \mathbf{M} + \mathbf{E}))(z - \mathbf{M} + \mathbf{E})^{-1} dz \right\| \\
&= \left\| \frac{1}{2\pi i} \oint_{\partial R} (z - \mathbf{M})^{-1} \mathbf{E} (z - \mathbf{M} + \mathbf{E})^{-1} dz \right\| \\
&\leq \frac{1}{2\pi} \oint_{\partial R} \left\| (z - \mathbf{M})^{-1} \mathbf{E} (z - \mathbf{M} + \mathbf{E})^{-1} \right\| dz \\
&\leq \frac{1}{2\pi} \text{length}(R) \cdot \|\mathbf{E}\| \cdot \sup_{z \in \partial R} \|(z - \mathbf{M})^{-1}\| \sup_{z \in \partial R} \|(z - \mathbf{M} + \mathbf{E})^{-1}\| \\
&\leq \frac{1}{2\pi} (6|\lambda_{\min}(\mathbf{M})| + 4) \cdot \epsilon_1 \frac{|\lambda_{\min}(\mathbf{M})|}{4\kappa_{\mathbf{V}}(\mathbf{M})} \cdot \frac{1}{|\lambda_{\min}(\mathbf{M})|} \cdot \frac{2}{|\lambda_{\min}(\mathbf{M})|} \\
&\leq \epsilon_1 \frac{4}{\pi \kappa_{\mathbf{V}}(\mathbf{M})|\lambda_{\min}(\mathbf{M})|}
\end{aligned}
$$

Since $|\lambda_{\min}(\mathbf{M})| \leq 2$, then if we set $\epsilon_1 \leq \epsilon \frac{|\lambda_{\min}(\mathbf{M})|\pi}{16}$ (which satisfies $\epsilon_1 < 1$), where ϵ is the desired final accuracy, this gives

$$\|\mathbf{P}_+(\mathbf{M}) - \mathbf{P}_+(\mathbf{M} + \mathbf{E})\| \leq \epsilon/2.$$

The same can be obtained for $\mathbf{P}_-$. Putting everything together, if $\|\mathbf{E}\| \leq \epsilon \frac{|\lambda_{\min}(\mathbf{M})|^2 \pi}{128 \kappa_{\mathbf{V}}(\mathbf{M})}$ for some $\epsilon \in (0,1)$ ensures that $\|\text{sgn}(\mathbf{M} + \mathbf{E}) - \text{sgn}(\mathbf{M})\| \leq \epsilon$. From Proposition A.2 $\kappa_{\mathbf{V}}(\mathbf{M}) \leq \sqrt{\kappa(\mathbf{S})}$, which gives the final bound. $\qquad\square$

We can now prove Proposition 2.1, which we restate for readability.

Proposition B.1 (Restatement of Proposition 2.1). *Let $\mathbf{H} \in \mathbb{H}^n$ with $\|\mathbf{H}\| \leq 1$, $\mathbf{S} \in \mathbb{H}_{++}^n$ with $\|\mathbf{S}^{-1}\| \leq 1$, $k \in [n-1]$ and $\epsilon \in (0,1)$. Let $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$ and $\text{gap}_k = \lambda_k - \lambda_{k+1}$, where $\lambda_1 \leq \ldots \leq \lambda_n$ are the generalized eigenvalues of the Hermitian definite pencil $(\mathbf{H}, \mathbf{S})$ and assume that we want to compute $\mathbf{\Pi}_k$ which is the true spectral projector associated with the k smallest eigenvalues. If we have access to*

$$\widetilde{\mu}_k \in \mu_k \pm \tfrac{1}{8} \text{gap}_k \quad \widetilde{\text{gap}}_k \in (1 \pm \tfrac{1}{8}) \text{gap}_k, \quad \widetilde{\kappa} \in [\kappa(\mathbf{S}), C\kappa(\mathbf{S})],$$

for some constant $C > 1$, then Algorithm 3 computes $\widetilde{\mathbf{\Pi}}_k \leftarrow \text{PURIFY}(\mathbf{H}, \mathbf{S}, \widetilde{\mu}_k, \widetilde{\text{gap}}_k, \widetilde{\kappa}, \epsilon)$ such that $\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\| \leq \epsilon$, in $O\left(T_{\text{MM}}(n)\left(\log(\frac{1}{\text{gap}_k}) + \log(\log(\frac{\kappa(\mathbf{S})}{\epsilon \, \text{gap}_k}))\right)\right)$ floating point operations using $O\left(\log(n) \log^3(\frac{1}{\text{gap}_k}) \log(\frac{\kappa(\mathbf{S})}{\epsilon \, \text{gap}_k})\right)$ bits of precision.

Proof. Using the notation of Algorithm 3, we have that $\widetilde{\mathbf{H}} = \mathbf{S}^{-1}\mathbf{H} + \mathbf{E}_1^{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{MM}}$. Let us initially set

$$\mathbf{u} = \epsilon_0 \frac{1}{\mu(n)\widetilde{\kappa}^{c_{\mathsf{INV}}\log(n)}} \le \epsilon_0 \frac{1}{\mu(n)\kappa(\mathbf{S})^{c_{\mathsf{INV}}\log(n)}}, \tag{6}$$

for some $\epsilon_0 \in (0, 1/8)$ (to be determined later). From Theorems A.3 and A.2, and from the assumption that $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\| \le 1$ we have that

$$\|\mathbf{E}_1^{\mathsf{INV}}\| \le \mathbf{u}\mu_{\mathsf{INV}}(n)\kappa(\mathbf{S})^{c_{\mathsf{INV}}\log(n)}\|\mathbf{S}^{-1}\| \le \epsilon_0,$$

$$\|\mathbf{E}_2^{\mathsf{MM}}\| \le \mathbf{u}\mu_{\mathsf{MM}}(n)\|\mathbf{H}\|\|\mathbf{S}_{\mathsf{INV}}\| \le \mathbf{u}\mu_{\mathsf{MM}}(n)\|\mathbf{H}\|\left(\|\mathbf{S}^{-1}\| + \|\mathbf{E}_1^{\mathsf{INV}}\|\right) \le \epsilon_0(1 + \epsilon_0) \le 2\epsilon_0.$$

We next have $\widetilde{\mathbf{M}} \leftarrow \widetilde{\mu}_k - \widetilde{\mathbf{H}} + \mathbf{E}_3^{(-)} = \widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H} + \mathbf{E}_1^{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{MM}} + \mathbf{E}_3^{(-)} = \widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H} + \mathbf{B}$, where $\mathbf{E}_3^{(-)}$ is a diagonal error matrix with

$$\|\mathbf{E}_3^{(-)}\| \le \mathbf{u}\|\widetilde{\mu}_k\|\|\widetilde{\mathbf{H}}\| \le \mathbf{u}\|\mathbf{S}^{-1}\mathbf{H} + \mathbf{E}_1^{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{MM}}\| \le \mathbf{u}(1 + \epsilon_0 + 2\epsilon_0) \le 4\mathbf{u} \ll 4\epsilon_0,$$

and $\mathbf{B} = \mathbf{E}_1^{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{MM}} + \mathbf{E}_3^{(-)}$. We can bound the norm of $\mathbf{B}$ as:

$$\|\mathbf{B}\| \le 7\epsilon_0.$$

Now we need to apply Lemma B.1 to argue that $\left\|\mathrm{sgn}(\mu_k - \mathbf{S}^{-1}\mathbf{H}) - \mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H} + \mathbf{B})\right\| \le \epsilon_1$, for some $\epsilon_1 \in (0, 1)$. To satisfy the requirements of the lemma, we need to ensure that $\|\mathbf{B}\| \le \epsilon_1 \frac{\pi|\lambda_{\min}^2(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})|}{128\sqrt{\kappa(\mathbf{S})}}$. This can be achieved by setting

$$\epsilon_0 = \epsilon_1 \frac{\pi|\lambda_{\min}^2(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})|}{7 \cdot 128\sqrt{\kappa(\mathbf{S})}}.$$

However, we do not know $|\lambda_{\min}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})|$ and $\kappa(\mathbf{S})$. To circumvent this, we can use the assumption that $\widetilde{\mu}_k$ is well-placed close to the center of gap_k, which implies that

$$|\lambda_{\min}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})| \ge \tfrac{3}{8}\mathrm{gap}_k \ge \tfrac{1}{3}\widetilde{\mathrm{gap}}_k,$$

and also use the assumption that $\kappa(\mathbf{S}) \le \widetilde{\kappa}$. This means that we can set

$$\epsilon_0 = \epsilon_1 \frac{\pi}{7 \cdot 128\sqrt{\widetilde{\kappa}}} \cdot \frac{1}{9}\widetilde{\mathrm{gap}}_k^2, \tag{7}$$

which implies the desired bound for $\mathbf{B}$:

$$\|\mathbf{B}\| \le 7 \cdot \epsilon_1 \frac{\pi}{7 \cdot 128\sqrt{\widetilde{\kappa}}} \cdot \frac{1}{9}\widetilde{\mathrm{gap}}_k^2 \le \epsilon_1 \frac{\pi|\lambda_{\min}^2(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})|}{128\sqrt{\kappa(\mathbf{S})}}.$$

We can now apply Lemma B.1 to argue that

$$\left\|\mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}) - \mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H} + \mathbf{B})\right\| \le \epsilon_1.$$

By assumption we know that $\widetilde{\mu}_k = \mu_k \pm \frac{1}{8}\mathrm{gap}_k$ which means that $\mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}) = \mathrm{sgn}(\mu_k - \mathbf{S}^{-1}\mathbf{H})$, and therefore

$$\left\|\mathrm{sgn}(\mu_k - \mathbf{S}^{-1}\mathbf{H}) - \mathrm{sgn}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H} + \mathbf{B})\right\| \le \epsilon_1.$$

To not interrupt the flow, we will skip for now the analysis of SGN and we will leave it for the end of the proof. From Theorem A.4 we know that the result returned by SGN satisfies $\|\widetilde{\mathbf{C}} - \mathrm{sgn}(\widetilde{\mathbf{M}})\| \le \epsilon_{\mathsf{SGN}}$, where we have full control over ϵ_{SGN} since it is passed as an argument to SGN, and therefore it can be set to $\epsilon_{\mathsf{SGN}} = \epsilon_1$. Denoting $\mathbf{C} = \mathrm{sgn}(\mu_k - \mathbf{S}^{-1}\mathbf{H})$ and $\mathbf{E_C} = \widetilde{\mathbf{C}} - \mathbf{C}$, we have

$$\left\|\widetilde{\mathbf{C}} - \mathrm{sgn}(\mu_k - \mathbf{S}^{-1}\mathbf{H})\right\| = \|\mathbf{E_C}\| \le \epsilon_1 + \epsilon_{\mathsf{SGN}} = 2\epsilon_1.$$

29

We now proceed to the line 5 of Algorithm 3, which computes $\widetilde{\Pi} \leftarrow \frac{1}{2}\left(1 + \widetilde{\mathbf{C}}\right) + \mathbf{E}_5^{(+)}$, where $\mathbf{E}_5^{(+)}$ is a diagonal error matrix with norm bounded by

$$\|\mathbf{E}_5^{(+)}\| \leq \mathbf{u}\|\widetilde{\mathbf{C}}\| \leq \mathbf{u}(1 + \epsilon_1 + \epsilon_{\mathsf{SGN}}) \leq 3\mathbf{u} \ll 3\epsilon_1.$$

Then we can write $\widetilde{\Pi} = \frac{1}{2}\left(1 + \mathbf{C} + \mathbf{E_C}\right) + \mathbf{E}_5^{(+)} = \Pi + \frac{1}{2}\mathbf{E_C} + \mathbf{E}_5^{(+)} = \Pi + \mathbf{E_\Pi}$, where $\Pi = \frac{1}{2}(1 + \mathbf{C})$ and $\mathbf{E_\Pi} = \frac{1}{2}\mathbf{E_C} + \mathbf{E}_5^{(+)}$. Combining with the above:

$$\|\mathbf{E_\Pi}\| \leq 4\epsilon_1,$$

in which case we can set $\epsilon_1 = \frac{\epsilon}{4}$, where ϵ is the desired accuracy, to guarantee forward error of at most ϵ for the spectral projector $\widetilde{\Pi}$.

Gathering all the requirements for the machine precision (except for SGN, which is detailed below) from Equations (6) and (7), and the assumption on $\widetilde{\kappa}$ and $\widetilde{\mathrm{gap}}_k$ it suffices to set

$$\mathbf{u} \leq \frac{\epsilon}{4}\frac{\pi}{7 \cdot 128\sqrt{\widetilde{\kappa}}} \cdot \frac{1}{9}\widetilde{\mathrm{gap}}_k^2 \frac{1}{\mu(n)\widetilde{\kappa}^{c_{\mathsf{INV}}}\log(n)},$$

which translates to

$$O\left(\log(1/\mathbf{u})\right) = O\left(\log\left(\frac{\sqrt{\widetilde{\kappa}}\mu(n)\widetilde{\kappa}^{c_{\mathsf{INV}}}\log(n)}{\epsilon\widetilde{\mathrm{gap}}_k^2}\right)\right) = O\left(\log(n)\log(\kappa(\mathbf{S})) + \log\left(\frac{n}{\epsilon\,\mathrm{gap}_k}\right)\right) \quad (8)$$

bits of precision.

We finally proceed with the analysis of SGN. To use Theorem A.4 for the convergence of SGN, we need to find the appropriate parameters $\alpha_{\mathsf{SGN}}, \eta_{\mathsf{SGN}}$ to call it. These parameters must be such that $\Lambda_{\eta_{\mathsf{SGN}}}(\widetilde{\mathbf{M}}) \subseteq \mathsf{C}_{\alpha_{\mathsf{SGN}}}$, and $\frac{99}{100} < \alpha_{\mathsf{SGN}} < 1$. From the properties of the pseudospectrum, specifically, from Proposition A.1 (i), we know that for any $\eta > 0$

$$\Lambda_\eta(\widetilde{\mathbf{M}}) = \Lambda_\eta(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H} + \mathbf{B}) \subseteq \Lambda_{\eta+\|\mathbf{B}\|}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}) \subseteq \Lambda_{\eta+B_{\max}}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}),$$

where $B_{\max}$ is the upper bound for $\|\mathbf{B}\|$ that we obtained above:

$$\|\mathbf{B}\| \leq \epsilon_1 \frac{\pi\widetilde{\mathrm{gap}}_k^2}{9 \cdot 128\sqrt{\widetilde{\kappa}}} := B_{\max}.$$

For $\eta_{\mathsf{SGN}} = B_{\max}$ and from Proposition A.1 (iii) we obtain

$$\begin{aligned}
\Lambda_{B_{\max}}(\widetilde{\mathbf{M}}) &\subseteq \Lambda_{2B_{\max}}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}) \\
&\subseteq \bigcup_i D\left(\widetilde{\mu}_k - \lambda_i(\mathbf{S}^{-1}\mathbf{H}), 2B_{\max}\kappa_{\mathbf{V}}(\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H})\right) \\
&\subseteq \bigcup_i D\left(\widetilde{\mu}_k - \lambda_i(\mathbf{S}^{-1}\mathbf{H}), 2\epsilon_1 \frac{\pi\widetilde{\mathrm{gap}}_k^2}{9 \cdot 128\sqrt{\widetilde{\kappa}}}\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H})\right).
\end{aligned}$$

where we used the fact that $\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H}) \leq \sqrt{\kappa(\mathbf{S})}$. We can get a rough upper bound

$$2\epsilon_1 \frac{\pi\widetilde{\mathrm{gap}}_k^2}{9 \cdot 128\sqrt{\widetilde{\kappa}}}\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H}) \leq \epsilon_1 \frac{2\pi \cdot 9^2}{9 \cdot 128 \cdot 8^2\sqrt{\kappa(\mathbf{S})}}\sqrt{\kappa(\mathbf{S})}\,\mathrm{gap}_k^2 \leq \epsilon_1 \frac{\mathrm{gap}_k^2}{128},$$

which gives

$$\Lambda_{B_{\max}}(\widetilde{\mathbf{M}}) \subseteq \bigcup_i D\left(\widetilde{\mu}_k - \lambda_i(\mathbf{S}^{-1}\mathbf{H}), \epsilon_1\frac{\mathrm{gap}_k^2}{128}\right).$$

It suffices to find the appropriate α_{SGN} such that the Apollonian circles $\mathsf{C}_{\alpha_{\mathsf{SGN}}}$ will contain the disks of the pseudospectrum. Since the smallest singular value of $\widetilde{\mu}_k - \mathbf{S}^{-1}\mathbf{H}$ is lower bounded by $\frac{3}{8}\mathrm{gap}_k$, the leftmost positive point where any pseudospectral disk intersects with the real axis is $\frac{3}{8}\mathrm{gap}_k - \epsilon_1\frac{\mathrm{gap}_k^2}{128} \geq \mathrm{gap}_k(\frac{3}{8} - \frac{1}{128}) = \mathrm{gap}_k\frac{47}{128} > \frac{\widetilde{\mathrm{gap}}_k}{4}$. Similarly for the negative halfplane, the rightmost negative point where a disk intersects with the real axis is $-\frac{\widetilde{\mathrm{gap}}_k}{4}$. Denoting by $\zeta = \frac{\widetilde{\mathrm{gap}}_k}{8}$,

it suffices to set $\alpha_{\mathsf{SGN}} = \frac{1-\zeta}{1+\zeta} = \frac{8-\widetilde{\mathrm{gap}}_k}{8+\widetilde{\mathrm{gap}}_k}$. Clearly, $\alpha_{\mathsf{SGN}} < 1$ for all $\infty > \mathrm{gap}_k > 0$. For the lower bound, to ensure $\alpha_{\mathsf{SGN}} > 99/100$ we need that $\widetilde{\mathrm{gap}}_k < \frac{8}{199}$. To achieve this we can simply scale the input matrices by a constant, since, by assumption, $\|\mathbf{S}^{-1}\mathbf{H}\| \leq 1$, and therefore $\mathrm{gap}_k \leq 2$. To conclude, we have all the required parameters to call SGN.

We can now go back to the complexity analysis. We are running

$$\widetilde{\mathbf{C}} \leftarrow \mathsf{SGN}\left(\widetilde{\mathbf{M}}, \alpha_{\mathsf{SGN}}, \eta_{\mathsf{SGN}}, \epsilon_{\mathsf{SGN}}\right),$$

where

$$\alpha_{\mathsf{SGN}} = \frac{8-\widetilde{\mathrm{gap}}_k}{8+\widetilde{\mathrm{gap}}_k}, \qquad \eta_{\mathsf{SGN}} = B_{\max} = \epsilon_1 \frac{\pi \widetilde{\mathrm{gap}}_k^2}{9 \cdot 128\sqrt{\widetilde{\kappa}}}, \qquad \epsilon_{\mathsf{SGN}} = \epsilon_1 = \frac{\epsilon}{4}.$$

We now invoke Theorem A.4. The number of iterations is bounded by

$$N = O\left(\log(\tfrac{1}{1-\alpha_{\mathsf{SGN}}}) + \log(\log(\tfrac{1}{\eta_{\mathsf{SGN}}\epsilon_{\mathsf{SGN}}}))\right)$$

$$= O\left(\log(\tfrac{1}{\widetilde{\mathrm{gap}}_k}) + \log(\log(\tfrac{\widetilde{\kappa}}{\epsilon\widetilde{\mathrm{gap}}_k}))\right)$$

$$= O\left(\log(\tfrac{1}{\mathrm{gap}_k}) + \log(\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k}))\right).$$

The number of required bits is

$$O\left(\log(n)\log^3(\tfrac{1}{1-\alpha_{\mathsf{SGN}}})\log(\tfrac{1}{\eta_{\mathsf{SGN}}\epsilon_{\mathsf{SGN}}})\right) = O\left(\log(n)\log^3(\tfrac{1}{\widetilde{\mathrm{gap}}_k})\log\left(\tfrac{\widetilde{\kappa}}{\epsilon\widetilde{\mathrm{gap}}_k}\right)\right)$$

$$= O\left(\log(n)\log^3(\tfrac{1}{\mathrm{gap}_k})\log\left(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k}\right)\right),$$

which dominates the previous bit bound of Equation 8.

It remains to bound the arithmetic complexity. There is a constant number of inversions, matrix multiplications, and scalar-matrix operations, which in total take at most $O(T_{\mathsf{MM}}(n))$ floating point operations. Thus, the dominant factor is the call to SGN, which amounts to

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{1}{\mathrm{gap}_k}) + \log(\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k}))\right)\right)$$

arithetic operations using the aforementioned number of bits. □

B.1 Spectral gaps with diagonalization

The straightforward approach to compute the desired spectral gaps is to iteratively compute $\mathbf{S}^{-1}\mathbf{H}$ and diagonalize it, until the eigenvalues are well approximated. If we compute $\mathbf{A} \leftarrow \mathsf{MM}(\mathsf{INV}(\mathbf{S}), \mathbf{H})$, then we can write $\mathbf{A} = \mathbf{S}^{-1}\mathbf{H} + \mathbf{E}$, for some error matrix $\mathbf{E}$. The next step is to approximate $\mathrm{gap}_k(\mathbf{S}^{-1}\mathbf{H})$ and $\mu_k(\mathbf{S}^{-1}\mathbf{H})$. For this one could use the recent state-of-the-art backward-approximate diagonalization algorithm of [15].

Theorem B.1 (EIG, imported Theorem 1.6 from [15]). *There exists a randomized algorithm* $\mathsf{EIG}(\mathbf{A}, \epsilon_{\mathsf{EIG}})$ *which takes any matrix* $\mathbf{A} \in \mathbb{C}^{n \times n}$ *with* $\|\mathbf{A}\| \leq 1$ *and a desired accuracy parameter* $\epsilon_{\mathsf{EIG}} > 0$ *as inputs and returns a diagonal* $\mathbf{D}$ *and an invertible matrix* $\mathbf{V}$ *such that*

$$\|\mathbf{A} - \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\| \leq \epsilon_{\mathsf{EIG}} \quad \text{and} \quad \kappa(\mathbf{V}) \leq 32n^{2.5}/\epsilon_{\mathsf{EIG}},$$

in

$$O\left(T_{\mathsf{MM}}(n)\log^2(\tfrac{n}{\epsilon_{\mathsf{EIG}}})\right)$$

arithmetic operations on a floating point machine with

$$O(\log^4(n/\epsilon_{\mathsf{EIG}})\log(n))$$

bits of precision, with probability at least $1 - 14/n$.

Applying EIG on $\mathbf{A}$ we can obtain a backward-approximate diagonalization. But we are not finished yet, since we are interested in each individual eigenvalue. To translate the backward error to a forward error for the eigenvalues, and, ultimately, the spectral gap, one can try to use Corollary 1.7 and Proposition 1.1 of [15]. However, this approach has two main limitations. First, it relies on simplicity of the spectrum, i.e., it assumes that the minimum gap between any pair of eigenvalues is larger than zero. This assumption is quite restrictive, since the desired gap might be well-defined even at the presence of other multiple eigenvalues. For example, in DFT applications it is not uncommon to have eigenenergies with algebraic multiplicity larger than one, and at the same time have a large band-gap that separates the occupied from the unoccupied orbitals. The second limitation is that the aforementioned Corollary 1.7 requires as an input parameter an actual over-estimate for the minimum eigenvalue gap. Even if such a gap exists, it is not described how to estimate it.

For diagonalizable matrices, we can leverage the following Corollary B.1, which is an immediate consequence of Kahan's inequality (Fact A.1), and it overcomes the aforementioned limitations.

Corollary B.1. *If* $\mathbf{X}$ *is diagonalizable and it has real eigenvalues then for any* $\mathbf{Z}$ *the following bound holds for the eigenvalues of* $\mathbf{X}$:

$$|\lambda_i(\mathbf{X}) - \lambda_i(\mathbf{Z})| \leq O(\log(n))\kappa_{\mathbf{V}}(\mathbf{X})\|\mathbf{X} - \mathbf{Z}\|.$$

Proof. Write $\mathbf{X} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}$ where $\mathbf{W}$ diagonalizes $\mathbf{X}$ and is chosen such that $\kappa(\mathbf{W}) = \kappa_{\mathbf{V}}(\mathbf{X})$. This is always possible since if $\mathbf{W}$ is any matrix that diagonalizes $\mathbf{X}$ then $\mathbf{\Lambda}$ is similar to $\mathbf{X}$ and since $\mathbf{X}$ has real eigenvalues then $\mathbf{\Lambda}$ has to be real, and therefore symmetric. Then we can write

$$\|\mathbf{\Lambda} - \mathbf{W}^{-1}\mathbf{Z}\mathbf{W}\| \leq \|\mathbf{W}\|\|\mathbf{W}^{-1}\|\|\mathbf{X} - \mathbf{Z}\| = \kappa(\mathbf{W})\|\mathbf{X} - \mathbf{Z}\|.$$

Since $\mathbf{\Lambda}$ is symmetric, and since $\mathbf{W}^{-1}\mathbf{Z}\mathbf{W}$ is similar to $\mathbf{Z}$ then from Kahan's inequality (Fact A.1)

$$\begin{aligned}|\lambda_i(\mathbf{X}) - \lambda_i(\mathbf{Z})| = |\lambda_i(\mathbf{\Lambda}) - \lambda_i(\mathbf{W}^{-1}\mathbf{Z}\mathbf{W})| &\leq O(\log(n))\|\mathbf{\Lambda} - \mathbf{W}^{-1}\mathbf{Z}\mathbf{W}\| \\ &\leq O(\log(n))\kappa(\mathbf{W})\|\mathbf{X} - \mathbf{Z}\| \\ &= O(\log(n))\kappa_{\mathbf{V}}(\mathbf{X})\|\mathbf{X} - \mathbf{Z}\|.\end{aligned}$$

$\square$

Now we need to use this to get a bound for the computed generalized eigenvalues after INV and EIG.

Proposition B.2 (EIG-based gap). *Given a definite pencil* $(\mathbf{H}, \mathbf{S})$ *with* $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\| \leq 1$, *we can compute* $\widetilde{\mathrm{gap}}_k \in (1 \pm \epsilon)\,\mathrm{gap}_k$ *and* $\widetilde{\mu}_k \in \mu_k \pm \epsilon\,\mathrm{gap}_k$ *by iteratively calling* INV, MM, *and* EIG, *using*

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{1}{\epsilon\,\mathrm{gap}_k})\log^2(\tfrac{n\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)$$

floating point operations using

$$O\left(\log^4(\tfrac{n\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\log(n)\right)$$

bits of precision with probability at least $1 - O(\tfrac{\log(1/\epsilon\,\mathrm{gap}_k)}{n})$.

Proof. Let $\mathbf{D}, \mathbf{V} \leftarrow \mathsf{EIG}(\mathbf{A}, \epsilon_{\mathsf{EIG}})$ be the solution returned by EIG when applied to $\mathbf{A} = \mathbf{S}^{-1}\mathbf{H} + \mathbf{E}$. Note that

$$\|\mathbf{S}^{-1}\mathbf{H} - \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\| = \|\mathbf{S}^{-1}\mathbf{H} + \mathbf{E} - \mathbf{E} - \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\| \leq \|\mathbf{S}^{-1}\mathbf{H} + \mathbf{E} - \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\| + \|\mathbf{E}\| \leq \epsilon_{\mathsf{EIG}} + \|\mathbf{E}\|.$$

From Corollary B.1, since $\mathbf{S}^{-1}\mathbf{H}$ is diagonalizable with real eigenvalues then we conclude that

$$\begin{aligned}|\lambda_i(\mathbf{S}^{-1}\mathbf{H}) - \mathbf{D}_{i,i}| = |\lambda_i(\mathbf{S}^{-1}\mathbf{H}) - \lambda_i(\mathbf{V}\mathbf{D}\mathbf{V}^{-1})| &\leq O(\log(n))\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H})\|\mathbf{S}^{-1}\mathbf{H} - \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\| \\ &\leq O(\log(n))\kappa_{\mathbf{V}}(\mathbf{S}^{-1}\mathbf{H})\left(\epsilon_{\mathsf{EIG}} + \|\mathbf{E}\|\right).\end{aligned}$$

We can tune the machine precision such that $\|\mathbf{E}\| = \epsilon_{\mathsf{EIG}} = \epsilon' \frac{1}{c\log(n)\sqrt{\kappa(\mathbf{S})}}$ for some chosen ϵ', and some global constant c, then finally

$$|\lambda_i(\mathbf{S}^{-1}\mathbf{H}) - \lambda_i(\mathbf{V}\mathbf{D}\mathbf{V}^{-1})| \leq \epsilon'.$$

32

We can now consider an iterative scheme, where we call EIG on $\mathbf{S}^{-1}\mathbf{H}$, halving ϵ' at each step. We need to keep halving ϵ' until it reaches $\epsilon' = \Theta(\epsilon\,\mathrm{gap}_k)$, in which case we have a total of $O(\log(\frac{1}{\epsilon\,\mathrm{gap}_k}))$ calls to EIG. In the worst case every call costs

$$O\left(T_{\mathsf{MM}}(n)\log^2(\tfrac{n}{\epsilon_{\mathsf{EIG}}})\right) = O\left(T_{\mathsf{MM}}(n)\log^2(\tfrac{n\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)$$

arithmetic operations using $O\left(\log^4(\frac{n\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\log(n)\right)$ bits (note that the bits required by MM and INV to achieve accuracy ϵ' are dominated by those required for EIG, and we therefore ignore them). Since we do not know gap_k, we can set the termination criterion to be $\epsilon' \approx \Theta(\epsilon\widetilde{\mathrm{gap}}_k)$, where $\widetilde{\mathrm{gap}}_k$ is the approximate gap that we obtain from EIG.

Each iteration succeeds with high probability $1 - 1/n$, in which case a union bound gives $1 - O(\log(1/(\epsilon\,\mathrm{gap}_k))/n$, which can potentially be improved to $1 - O(1/n)$ without impacting the complexity, but we do not expand further.

A subtle detail in the analysis above is that we need an estimate for $\kappa(\mathbf{S})$ in order to be able to use INV and to set the machine precision. Since $\kappa(\mathbf{S})$ is generally unknown, we need to compute it. We will later show in Appendix E how to compute the condition number quickly.

When $\mathbf{S} = \mathbf{I}$, we can avoid the inversion and the computation of $\kappa(\mathbf{S})$. The arithmetic complexity and the bit requirement are the same by setting $\kappa(\mathbf{S}) = 1$. $\qquad\square$

Remark B.1. *We highlight that a similar result can be obtained by using the more recent pencil diagonalization algorithm of [41]. The authors the latter mention that their algorithm should be "more numerically stable" than* EIG *of [15], since it uses an inverse-free iteration internally, and they do provide strong theoretical and experimental evidence for this statement. However, as there is no formal, end-to-end proof for the bit complexity at the time of this writing, we choose to compare against* EIG *of [15]. Note that, any improvements on the bit requirements for the matrix sign function, directly provide the same improvements for our main Theorems 1.1 and 3.1.*

C Analysis of Cholesky

This section is devoted to the analysis of Algorithm 2 and the proof of Theorem 1.2. Let $\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{C} \end{pmatrix}$ be a Hermitian matrix. It can be factorized in the form

$$\mathbf{M} = \begin{pmatrix} \mathbf{I} & \\ \mathbf{B}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \\ & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B}^* \\ & \mathbf{I} \end{pmatrix} = \mathbf{W}\mathbf{Y}\mathbf{W}^*, \tag{9}$$

where $\mathbf{S} = \mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*$ is the Schur complement and

$$\mathbf{W} = \begin{pmatrix} \mathbf{I} & \\ \mathbf{B}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix}, \qquad \mathbf{Y} = \begin{pmatrix} \mathbf{A} & \\ & \mathbf{S} \end{pmatrix}.$$

Proposition C.1. *Let $\mathbf{M} \in \mathbb{H}_{++}^n$ and consider the partitioning $\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{C} \end{pmatrix}$. The following hold*

(i) *$\mathbf{A}, \mathbf{C}$ and the Schur complement $\mathbf{S} = \mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*$ are all positive definite;*

(ii) *For all $\mathbf{X} \in \{\mathbf{A}, \mathbf{C}, \mathbf{S}\}$ we have that $\|\mathbf{X}\| \le \|\mathbf{M}\|$ and $\|\mathbf{X}^{-1}\| \le \|\mathbf{M}^{-1}\|$;*

(iii) *$\|\mathbf{B}\| \le \|\mathbf{M}\|/2$.*

Proof. It is easy to see that $\mathbf{A}$ is positive definite: since $\mathbf{M} \succ 0$, then it must also hold that $\mathbf{A} \succ 0$ since the quadratic form $\mathbf{x}^*\mathbf{A}\mathbf{x}$ can be written as $\mathbf{y}^*\mathbf{M}\mathbf{y}$ for some vector $\mathbf{y}$. For the Schur complement we recall the factorized form of Equation (9). Consider the quadratic form $\mathbf{x}^*(\mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*)\mathbf{x}$. Let $\mathbf{y} = \begin{pmatrix} -\mathbf{A}^{-1}\mathbf{B}^*\mathbf{x} \\ \mathbf{x} \end{pmatrix}$. Then $\mathbf{y}^*\mathbf{M}\mathbf{y} = \mathbf{x}^*(\mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*)\mathbf{x}$, which means that every quadratic form for the Schur complement can be written as a quadratic form for the matrix $\mathbf{M}$, and therefore they are both positive definite.

33

For the spectral norm bounds, since $\mathbf{A}$ and $\mathbf{C}$ are both positive definite then their norms are equal to the largest absolute eigenvalues. From the variational characterizaiton of eigenvalues and the discussion above it is easy to see that $\|\mathbf{A}\| \leq \|\mathbf{M}\|$. For the Schur complement, let $\mathbf{z}$ be the eigenvector such that $\|\mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*\| = \mathbf{z}^*(\mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*)\mathbf{z}$. Then

$$\mathbf{z}^*(\mathbf{C} - \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*)\mathbf{z} = \mathbf{z}^*\mathbf{C}\mathbf{z} - \mathbf{z}^*\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*\mathbf{z} \leq \mathbf{z}^*\mathbf{C}\mathbf{z} \leq \|\mathbf{M}\|,$$

where we used the fact that $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*$ is positive semi-definite (since $\mathbf{A}^{-1}$ is positive definite) and therefore for all $\mathbf{x}$

$$\mathbf{x}^*\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^*\mathbf{x} \geq 0,$$

where equality with zero holds only when $\mathbf{z} \in \ker(\mathbf{B}^*)$.

We finally prove the bound for $\|\mathbf{B}\|$. Let $\mathbf{u}$ be the top left singular vector of $\mathbf{B}$ and $\mathbf{v}$ be the top right singular vector. Specifically, if $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$ is the economy SVD of $\mathbf{B}$, then $\mathbf{u}$ is the first column of $\mathbf{U}$ and $\mathbf{v}^*$ is the first row of $\mathbf{V}^*$. Then $\mathbf{u}^*\mathbf{B}\mathbf{v} = \sigma_{\max}(\mathbf{B})\mathbf{v}^*\mathbf{v} = (\mathbf{B})\mathbf{u}^*\mathbf{u} = \|\mathbf{B}\|$. Consider the vector $\mathbf{z} = \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}$. Then

$$\|\mathbf{M}\| \geq \mathbf{z}^*\mathbf{M}\mathbf{z} = (\mathbf{v}^* \quad \mathbf{u}^*)\begin{pmatrix} \mathbf{A} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{C} \end{pmatrix}\begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix} = \mathbf{v}^*\mathbf{A}\mathbf{v} + 2\mathbf{u}^*\mathbf{B}\mathbf{v} + \mathbf{u}^*\mathbf{C}\mathbf{u} \geq 2\mathbf{u}^*\mathbf{B}\mathbf{v} = 2\|\mathbf{B}\|,$$

where we used the fact that $\mathbf{A}, \mathbf{C}$ are positive-definite. We can then obtain bounds for $\|\mathbf{C}^{-1}\|, \|\mathbf{A}^{-1}\|$, and $\|\mathbf{S}^{-1}\|$, by observing that $\sigma_{\min}(\mathbf{X}) = \min_{\|\mathbf{z}\|=1} \mathbf{z}^*\mathbf{X}\mathbf{z}$ and lower bound it by $\sigma_{\min}(\mathbf{M})$ using similar argumnents. $\qquad\square$

Since $\mathbf{A}$ and $\mathbf{S}$ are Hermitian and positive-definite, we can recursively compute their Cholesky factors. This gives rise to the recursive Algorithm 2. The complexity is as follows.

Proposition C.2. *Algorithm 2 requires $O(T_{\mathsf{MM}}(n))$ arithmetic operations.*

Proof. Let $T(n)$ be the number of operations executed for a matrix of size $n \times n$. Steps 4 and 8 require time $T(n/2)$. Step 5 requires time $T_{\mathsf{MM}}(n/2) + T_{\mathsf{INV}}(n/2)$, and step 6 requires $T_{\mathsf{MM}}(n/2)$. The Schur complement in Step 7 requires $T_{\mathsf{MM}}(n/2) + T_{\mathsf{HERM}}(n/2) + (n/2)^2 = T_{\mathsf{MM}}(n/2) + 2(n/2)^2$. This becomes

$$T(n) = 2T(\tfrac{n}{2}) + 3T_{\mathsf{MM}}(\tfrac{n}{2}) + T_{\mathsf{INV}}(\tfrac{n}{2}) + 2(\tfrac{n}{2})^2 \leq 2T(\tfrac{n}{2}) + O(T_{\mathsf{MM}}(\tfrac{n}{2})) \leq O(T_{\mathsf{MM}}(n)).$$

$\qquad\square$

C.1 Error analysis

The goal is to show that $\mathsf{CHOLESKY}(\mathbf{M})$ will return a backward-approximate Cholesky factor $\mathbf{L}$ such that the error

$$\|\mathbf{L}\mathbf{L}^* - \mathbf{A}\| \leq f(\mathbf{u}, \|\mathbf{M}\|, \|\mathbf{M}^{-1}\|, n),$$

where f is some function, and use this bound to argue about the number of bits that are required to achieve it. Recall that

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{C} \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} \mathbf{L}_{11} & \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix}, \quad \mathbf{L}\mathbf{L}^* = \begin{pmatrix} \mathbf{L}_{11}\mathbf{L}_{11}^* & \mathbf{L}_{11}\mathbf{L}_{21}^* \\ \mathbf{L}_{21}\mathbf{L}_{11}^* & \mathbf{L}_{21}\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* \end{pmatrix}.$$

It is easy to see that

$$\|\mathbf{L}\mathbf{L}^* - \mathbf{M}\| = \left\|\begin{pmatrix} \mathbf{L}_{11}\mathbf{L}_{11}^* & \mathbf{L}_{11}\mathbf{L}_{21}^* \\ \mathbf{L}_{21}\mathbf{L}_{11}^* & \mathbf{L}_{21}\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* \end{pmatrix} - \begin{pmatrix} \mathbf{A} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{C} \end{pmatrix}\right\|$$
$$\leq \|\mathbf{L}_{11}\mathbf{L}_{11}^* - \mathbf{A}\| + \|\mathbf{L}_{21}\mathbf{L}_{11}^* - \mathbf{B}\| + \|\mathbf{L}_{21}\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* - \mathbf{C}\|, \qquad (10)$$

which means that it suffices to bound the individual terms in the sum. The first term is the error of the first recursive call. Following the notation of [39], let us denote $\mathbf{err}(n)$ the norm-wise error of the algorithm for size n. We can then write

$$\|\mathbf{L}_{11}\mathbf{L}_{11}^* - \mathbf{A}\| \leq \|\mathbf{E}_0^{\mathsf{CH}}\| \leq \mathbf{err}(n/2). \qquad (11)$$

To simplify the various inequalities in the proofs we use the global bound μ from Remark A.1, where all the terms $\mu_{\mathsf{MM}}(n)$ and $\mu_{\mathsf{INV}}(n)$ are bounded by $\mu(n)$. Finally, we also define

$$\mathcal{E}_1(\mathbf{M}, n) := \mu(n) \cdot \kappa(\mathbf{M}),$$
$$\mathcal{E}_{\mathsf{INV}}(\mathbf{M}, n) := \mu(n) \cdot \kappa(\mathbf{M})^{c_{\mathsf{INV}} \log(n)}. \tag{12}$$

We can now get a first expression for the bounds of the terms in Equation (10).

Lemma C.1. *In Algorithm 2, the blocks* $\mathbf{L}_{11}, \mathbf{L}_{21}, \mathbf{L}_{22}$ *of the returned matrix* $\mathbf{L}$ *satisfy:*

$$\|\mathbf{L}_{11}\mathbf{L}_{11}^* - \mathbf{A}\| \leq \mathsf{err}(n/2),$$
$$\|\mathbf{L}_{21}\mathbf{L}_{11}^* - \mathbf{B}\| \leq (\kappa(\mathbf{M}) + \|\mathbf{E}_{\mathbf{BA}i}\|)\,\mathsf{err}(n/2) + \|\mathbf{E}_{\mathbf{BA}i}\|\|\mathbf{M}\| + \|\mathbf{E}_3^{\mathsf{MM}}\|\|\mathbf{L}_{11}^*\|,$$
$$\|\mathbf{L}_{21}\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* - \mathbf{C}\| \leq 2\kappa(\mathbf{M})^2 \cdot \mathsf{err}(n/2) + 2\kappa(\mathbf{M})\|\mathbf{E}_{\mathbf{L}_{21}}\|\|\mathbf{L}_{11}\| + \|\mathbf{E}_{\mathbf{L}_{21}}\|^2 + \|\mathbf{E}_{\mathbf{S}}\|.$$

Proof. The error for $\mathbf{L}_{11}\mathbf{L}_{11}^*$ was already described. For the second inequality, we first expand $\mathbf{L}_{21}\mathbf{L}_{11}^*$ as follows:

$$\begin{aligned}
\mathbf{L}_{21}\mathbf{L}_{11}^* &= (\mathbf{BA}^{-1} + \mathbf{E}_{\mathbf{BA}i})\mathbf{L}_{11}\mathbf{L}_{11}^* + \mathbf{E}_3^{\mathsf{MM}}\mathbf{L}_{11}^* \\
&= (\mathbf{BA}^{-1} + \mathbf{E}_{\mathbf{BA}i})(\mathbf{A} + \mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}) + \mathbf{E}_3^{\mathsf{MM}}\mathbf{L}_{11}^* \\
&= \mathbf{B} + \mathbf{BA}^{-1}\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}} + \mathbf{E}_{\mathbf{BA}i}\mathbf{A} + \mathbf{E}_{\mathbf{BA}i}\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}} + \mathbf{E}_3^{\mathsf{MM}}\mathbf{L}_{11}^*.
\end{aligned}$$

Then

$$\begin{aligned}
\|\mathbf{L}_{21}\mathbf{L}_{11}^* - \mathbf{B}\| &= \left\|\mathbf{BA}^{-1}\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}} + \mathbf{E}_{\mathbf{BA}i}\mathbf{A} + \mathbf{E}_{\mathbf{BA}i}\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}} + \mathbf{E}_3^{\mathsf{MM}}\mathbf{L}_{11}^*\right\| \\
&\leq \|\mathbf{B}\|\|\mathbf{A}^{-1}\|\|\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}\| + \|\mathbf{E}_{\mathbf{BA}i}\|\|\mathbf{A}\| + \|\mathbf{E}_{\mathbf{BA}i}\|\|\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}\| + \|\mathbf{E}_3^{\mathsf{MM}}\|\|\mathbf{L}_{11}^*\| \\
&\leq \kappa(\mathbf{M})\mathsf{err}(n/2) + \|\mathbf{E}_{\mathbf{BA}i}\|\|\mathbf{M}\| + \|\mathbf{E}_{\mathbf{BA}i}\|\mathsf{err}(n/2) + \|\mathbf{E}_3^{\mathsf{MM}}\|\|\mathbf{L}_{11}^*\|.
\end{aligned}$$

With a similar procedure we can derive the third bound. We first write $\mathbf{L}_{21} = \mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{E}_{\mathbf{L}_{21}}$ where $\mathbf{E}_{\mathbf{L}_{21}}$ contains several error matrices from previous computations. Then

$$\begin{aligned}
\mathbf{L}_{21}\mathbf{L}_{21}^* &= \left(\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{E}_{\mathbf{L}_{21}}\right)\left(\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{E}_{\mathbf{L}_{21}}\right)^* \\
&= \left(\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{E}_{\mathbf{L}_{21}}\right)\left(\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{E}_{\mathbf{L}_{21}}\right)^* \\
&= \mathbf{BA}^{-1}\mathbf{L}_{11}\mathbf{L}_{11}^*\mathbf{A}^{-1}\mathbf{B}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{L}_{11}^*\mathbf{A}^{-1}\mathbf{B}^*\mathbf{E}_{\mathbf{L}_{21}}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{E}_{\mathbf{L}_{21}}^* \\
&= \mathbf{BA}^{-1}(\mathbf{A} + \mathbf{E}_{\mathbf{A}}^{\mathsf{CH}})\mathbf{A}^{-1}\mathbf{B}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{L}_{11}^*\mathbf{A}^{-1}\mathbf{B}^*\mathbf{E}_{\mathbf{L}_{21}}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{E}_{\mathbf{L}_{21}}^* \\
&= \mathbf{BA}^{-1}\mathbf{B}^* + \mathbf{BA}^{-1}\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}\mathbf{A}^{-1}\mathbf{B}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{L}_{11}^*\mathbf{A}^{-1}\mathbf{B}^*\mathbf{E}_{\mathbf{L}_{21}}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{E}_{\mathbf{L}_{21}}^*.
\end{aligned}$$

Similarly

$$\begin{aligned}
\mathbf{L}_{22}\mathbf{L}_{22}^* &= \widetilde{\mathbf{S}} + \mathbf{E}_{\widetilde{\mathbf{S}}}^{\mathsf{CH}} \\
&= \mathbf{S} + \mathbf{E}_{\mathbf{S}} + \mathbf{E}_{\widetilde{\mathbf{S}}}^{\mathsf{CH}} \\
&= \mathbf{C} - \mathbf{BA}^{-1}\mathbf{B}^* + \mathbf{E}_{\mathbf{S}} + \mathbf{E}_{\widetilde{\mathbf{S}}}^{\mathsf{CH}}.
\end{aligned}$$

Then

$$\begin{aligned}
\|\mathbf{L}_{21}&\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* - \mathbf{C}\| \\
&= \left\|\mathbf{BA}^{-1}\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}\mathbf{A}^{-1}\mathbf{B}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{BA}^{-1}\mathbf{L}_{11} + \mathbf{L}_{11}^*\mathbf{A}^{-1}\mathbf{B}^*\mathbf{E}_{\mathbf{L}_{21}}^* + \mathbf{E}_{\mathbf{L}_{21}}\mathbf{E}_{\mathbf{L}_{21}}^* + \mathbf{E}_{\mathbf{S}} + \mathbf{E}_{\widetilde{\mathbf{S}}}^{\mathsf{CH}}\right\| \\
&\leq \|\mathbf{BA}^{-1}\|^2\|\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}\| + 2\|\mathbf{E}_{\mathbf{L}_{21}}\|\|\mathbf{BA}^{-1}\|\|\mathbf{L}_{11}\| + \|\mathbf{E}_{\mathbf{L}_{21}}\|^2 + \|\mathbf{E}_{\mathbf{S}}\| + \|\mathbf{E}_{\widetilde{\mathbf{S}}}^{\mathsf{CH}}\| \\
&\leq \kappa(\mathbf{M})^2 \cdot \mathsf{err}(n/2) + 2\kappa(\mathbf{M})\|\mathbf{E}_{\mathbf{L}_{21}}\|\|\mathbf{L}_{11}\| + \|\mathbf{E}_{\mathbf{L}_{21}}\|^2 + \|\mathbf{E}_{\mathbf{S}}\| + \mathsf{err}(n/2) \\
&\leq 2\kappa(\mathbf{M})^2 \cdot \mathsf{err}(n/2) + 2\kappa(\mathbf{M})\|\mathbf{E}_{\mathbf{L}_{21}}\|\|\mathbf{L}_{11}\| + \|\mathbf{E}_{\mathbf{L}_{21}}\|^2 + \|\mathbf{E}_{\mathbf{S}}\|.
\end{aligned}$$

$\square$

With this, the task reduces to finding appropriate bounds for the norms of the matrices $\mathbf{E}_3^{\mathsf{MM}}, \mathbf{E}_{\mathbf{BA}i}, \mathbf{E}_{\mathbf{S}}$ and $\mathbf{E}_{\mathbf{L}_{21}}$. We first derive bounds for the norms of the error matrices $\mathbf{E}_{(\cdot)}$.

Lemma C.2. *In Algorithm 2, the following bounds hold for the error matrices*

(i) $\left\|\mathbf{E}_1^{\mathsf{INV}}\right\| \leq \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \|\mathbf{M}^{-1}\|,$

(ii) $\left\|\mathbf{E}_2^{\mathsf{MM}}\right\| \leq \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right),$

(iii) $\|\mathbf{E}_{\mathbf{BA}i}\| \leq \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \cdot \kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right),$

(iv) $\left\|\mathbf{E}_3^{\mathsf{MM}}\right\| \leq \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{L}_{11}\|,$

(v) $\left\|\mathbf{E}_4^{\mathsf{MM}}\right\| \leq \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{M}\|,$

(vi) $\left\|\mathbf{E}_5^{\mathsf{SUB}}\right\| \leq \mathbf{u} \cdot \sqrt{n/2} \cdot \|\mathbf{M}\| \cdot \left(1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right)^2\right),$

(vii) $\|\mathbf{E}_{\mathbf{L}_{21}}\| \leq \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \cdot \left[\kappa(\mathbf{M}) + \left(2 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right)^2\right] \cdot \|\mathbf{L}_{11}\|,$

(viii) $\|\mathbf{E}_{\mathbf{S}}\| \leq \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \cdot \left(2\kappa(\mathbf{M}) + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right)^2 \left(3\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\right)\right).$

Proof. Each term is bounded as follows.

(i) For $\|\mathbf{E}_1^{\mathsf{INV}}\|$ we have:

$$\begin{aligned}
\|\mathbf{E}_1^{\mathsf{INV}}\| &\leq \mu_{\mathsf{INV}}(n/2) \cdot \mathbf{u} \cdot \kappa(\mathbf{A})^{c_{\mathsf{INV}} \log \frac{n}{2}} \|\mathbf{A}^{-1}\| && (\text{...from Thm. A.3}) \\
&\leq \mu(n/2) \cdot \mathbf{u} \cdot \kappa(\mathbf{A})^{c_{\mathsf{INV}} \log \frac{n}{2}} \|\mathbf{M}^{-1}\| \\
&= \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \|\mathbf{M}^{-1}\|.
\end{aligned}$$

(ii) Similarly, for $\|\mathbf{E}_2^{\mathsf{MM}}\|$:

$$\begin{aligned}
\|\mathbf{E}_2^{\mathsf{MM}}\| &\leq \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u}\|\mathbf{B}\|\|\mathbf{A}^{-1} + \mathbf{E}_1^{\mathsf{INV}}\| && (\text{...from Thm. A.2}) \\
&\leq \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u}\|\mathbf{B}\|\|\mathbf{A}^{-1}\| + \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u}\|\mathbf{B}\|\|\mathbf{E}_1^{\mathsf{INV}}\|. \\
&\leq \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u} \cdot \|\mathbf{M}\|\|\mathbf{M}^{-1}\| + \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \|\mathbf{M}^{-1}\| \\[4pt]
&\leq \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right) + \mathbf{u}^2 \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right) \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \\
&= \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right).
\end{aligned}$$

(iii) Using these two inequalities we can bound the norm of $\mathbf{E}_{\mathbf{BA}i}$:

$$\begin{aligned}
\|\mathbf{E}_{\mathbf{BA}i}\| &= \|\mathbf{B}\mathbf{E}_1^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{MM}}\| \\
&\leq \|\mathbf{B}\|\|\mathbf{E}_1^{\mathsf{INV}}\| + \|\mathbf{E}_2^{\mathsf{MM}}\| \\
&\leq \|\mathbf{M}\| \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) \|\mathbf{M}^{-1}\| + \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right) \\
&\leq \kappa(\mathbf{M}) \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right) + \mathbf{u} \cdot \mathcal{E}_1\left(\frac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\frac{n}{2}, \mathbf{M}\right)\right).
\end{aligned}$$

(iv) Next is the error term of the matrix multiplication between the result of $\mathsf{MM}(\mathbf{B}, \mathsf{INV}(A))$ and $\mathbf{L}_{11}$. Expanding Theorem A.2 gives

$$\|\mathbf{E}_3^{\mathsf{MM}}\| \leq \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u} \cdot \|\mathbf{L}_{11}\| \|\mathbf{BA}i\|.$$

It suffices to bound $\|\mathbf{BA}i\| = \left\|\mathbf{B}\left(\mathbf{A}^{-1} + \mathbf{E}_1^{\mathsf{INV}}\right) + \mathbf{E}_2^{\mathsf{MM}}\right\| \leq \|\mathbf{BA}^{-1}\| + \|\mathbf{BE}_1^{\mathsf{INV}}\| + \|\mathbf{E}_2^{\mathsf{MM}}\|$. Using Proposition C.1, the triangle inequality, and the previous bounds, the two

36

unknown norms in the sum are bounded as follows

$$\|\mathbf{B}\mathbf{A}^{-1}\| \le \|\mathbf{B}\|\|\mathbf{A}^{-1}\| \le \|\mathbf{M}\|\|\mathbf{M}^{-1}\| = \kappa(\mathbf{M}),$$

$$\|\mathbf{B}\mathbf{E}_1^{\mathsf{INV}}\| \le \|\mathbf{B}\|\|\mathbf{E}_1^{\mathsf{INV}}\|$$

$$\le \|\mathbf{M}\| \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\|\mathbf{M}^{-1}\|$$

$$= \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\kappa(\mathbf{M}).$$

This gives

$$\|\mathbf{B}\mathbf{A}i\| \le \kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)$$

$$= \kappa(\mathbf{M})\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)$$

$$\le \kappa(\mathbf{M})\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2,$$

where in the last inequality we simplified $\left(1 + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \le \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)$. Then

$$\|\mathbf{E}_3^{\mathsf{MM}}\| \le \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u} \cdot \|\mathbf{L}_{11}\| \cdot \kappa(\mathbf{M})\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)$$

$$\le \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{L}_{11}\|,$$

concluding the bound for $\mathbf{E}_3^{\mathsf{MM}}$.

(v) The norm of $\mathbf{E}_4^{\mathsf{MM}}$ is very similar to that of $\mathbf{E}_3^{\mathsf{MM}}$ since they both involve a multiplication with $\mathbf{B}\mathbf{A}i$:

$$\|\mathbf{E}_4^{\mathsf{MM}}\| \le c_{\mathsf{HERM}} \log(n/2) \cdot \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u} \cdot \|\mathbf{B}\mathbf{A}i\|\|\mathbf{B}\| \qquad (\text{...from Corollary A.2})$$

$$\le c_{\mathsf{HERM}} \log(n/2) \cdot \mu_{\mathsf{MM}}(n/2) \cdot \mathbf{u} \cdot \kappa(\mathbf{M})\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{M}\|$$

$$\le \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{M}\|.$$

(vi) Next is $\mathbf{E}_5^{\mathsf{SUB}}$. From Equation (5):

$$\|\mathbf{E}_5^{\mathsf{SUB}}\| \le \mathbf{u}\sqrt{n/2}\left\|\mathbf{C} - \left((\mathbf{B}\mathbf{A}i)\mathbf{B}^* + \mathbf{E}_4^{\mathsf{MM}}\right)\right\|$$

$$\le \mathbf{u}\sqrt{n/2}\left(\|\mathbf{C}\| + \|(\mathbf{B}\mathbf{A}i)\mathbf{B}^*\| + \|\mathbf{E}_4^{\mathsf{MM}}\|\right\|$$

$$\le \mathbf{u}\sqrt{n/2}\left(\|\mathbf{M}\| + \|(\mathbf{B}\mathbf{A}i)\mathbf{B}^*\| + \|\mathbf{E}_4^{\mathsf{MM}}\|\right)$$

$$\le \mathbf{u}\sqrt{n/2}\left(\|\mathbf{M}\| + \kappa(\mathbf{M})\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{M}\| + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{M}\|\right)$$

$$= \mathbf{u}\sqrt{n/2} \cdot \|\mathbf{M}\| \cdot \left(1 + \kappa(\mathbf{M})\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right)$$

$$= \mathbf{u}\sqrt{n/2} \cdot \|\mathbf{M}\| \cdot \left(1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right).$$

(vii) The next term is $\mathbf{E}_{\mathbf{L}_{21}}$, which we can bound as follows:

$$\|\mathbf{E}_{\mathbf{L}_{21}}\| = \|\mathbf{E}_{\mathbf{B}\mathbf{A}i}\mathbf{L}_{11} + \mathbf{E}_3^{\mathsf{MM}}\|$$

$$\le \|\mathbf{E}_{\mathbf{B}\mathbf{A}i}\| \cdot \|\mathbf{L}_{11}\| + \|\mathbf{E}_3^{\mathsf{MM}}\|$$

$$\le \left[\kappa(\mathbf{M}) \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\right] \cdot \|\mathbf{L}_{11}\| + \dots$$

$$\dots + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{L}_{11}\|$$

$$= \left[\kappa(\mathbf{M}) \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \cdot \left(2 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\right] \cdot \|\mathbf{L}_{11}\|$$

$$\le \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left[\kappa(\mathbf{M}) + \left(2 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right] \cdot \|\mathbf{L}_{11}\|.$$

(viii) The final and most involved error term is the norm of $\mathbf{E}_{\mathbf{S}}$. Recall that from Line 7 of Algorithm 2 the matrix can be written as

$$\mathbf{E}_{\mathbf{S}} = \mathbf{B}\mathbf{E}_1^{\mathsf{INV}}\mathbf{B}^* + \mathbf{E}_2^{\mathsf{MM}}\mathbf{B}^* + \mathbf{E}_4^{\mathsf{MM}} + \mathbf{E}_5^{\mathsf{SUB}}.$$

Once more, the norm of each term is bounded separately.

$$\begin{aligned}
\|\mathbf{B}\mathbf{E}_1^{\mathsf{INV}}\mathbf{B}^*\| &\leq \|\mathbf{B}\|\|\mathbf{E}_1^{\mathsf{INV}}\|\|\mathbf{B}^*\| \\
&\leq \|\mathbf{M}\|^2 \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\|\mathbf{M}^{-1}\| \\
&= \kappa(\mathbf{M}) \cdot \|\mathbf{M}\| \cdot \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right).
\end{aligned}$$

Similarly,

$$\|\mathbf{E}_2^{\mathsf{MM}}\mathbf{B}^*\| \leq \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \cdot \|\mathbf{M}\|.$$

The final bound for $\mathbf{E_S}$ is given by the sum of the four bounds that were derived.

$$\begin{aligned}
\|\mathbf{E_S}\| &= \|\mathbf{B}\mathbf{E}_1^{\mathsf{INV}}\mathbf{B}^* + \mathbf{E}_2^{\mathsf{MM}}\mathbf{B}^* + \mathbf{E}_4^{\mathsf{MM}} + \mathbf{E}_5^{\mathsf{SUB}}\| \\
&\leq \|\mathbf{B}\mathbf{E}_1^{\mathsf{INV}}\mathbf{B}^*\| + \|\mathbf{E}_2^{\mathsf{MM}}\mathbf{B}^*\| + \|\mathbf{E}_4^{\mathsf{MM}}\| + \|\mathbf{E}_5^{\mathsf{SUB}}\| \\
&\leq \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \kappa(\mathbf{M}) \cdot \|\mathbf{M}\| \dots \\
&\qquad \dots + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \cdot \|\mathbf{M}\| \\
&\qquad \dots + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{M}\| \\
&\qquad \dots + \mathbf{u}\sqrt{n/2} \cdot \|\mathbf{M}\| \cdot \left(1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right) \\
&= \mathbf{u} \cdot \|\mathbf{M}\| \cdot \left[\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \kappa(\mathbf{M}) \dots\right. \\
&\qquad \dots + \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \\
&\qquad \dots + \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \\
&\qquad \left.\dots + \sqrt{n/2}\left(1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right)\right] \\
&= \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left[\kappa(\mathbf{M}) + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \dots\right. \\
&\qquad \dots + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \\
&\qquad \left.\dots + \left(1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right)\right] \\
&= \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left[\kappa(\mathbf{M}) + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right) \dots\right. \\
&\qquad \dots + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \\
&\qquad \left.\dots + \left(1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right)\right] \\
&\leq \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left[\kappa(\mathbf{M}) + 1 + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\left(1 + 1 + \left(\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\right)\right] \\
&\leq \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left[2\kappa(\mathbf{M}) + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\left(3\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\right].
\end{aligned}$$

$$\square$$

C.2 Maintaining positive-definiteness, norms, and condition numbers throughout the recursion

Given the bounds of Lemma C.2, we can now calculate the appropriate machine precision $\mathbf{u}$ and the corresponding number of bits such that the algorithm will not break down due to loss of positive-definiteness of the submatrices. Hereafter, we will denote by $\mathbf{M}$ the matrix that is passed as an argument in any of the recursive calls of Algorithm 2, and $\mathbf{M}_0$ will denote the original matrix that needs to be factorized.

Lemma C.3. *Let $\mathbf{M}_0 \in \mathbb{H}^n_{++}$ be a matrix that is factorized using Algorithm 2 and κ be its condition number. Then there exist constants $c_1, c_2, c_3, c_4 \geq 1$, such that if $n > c_1$ and*

$$\mathbf{u} \leq \mathbf{u}_{++} := \frac{1}{c_2 \cdot n^{c_3} \cdot \kappa^{c_4 \log n}},$$

then every matrix that is constructed in Line 7 during the recursion, that is, every Schur complement $\widetilde{\mathbf{S}}$ and every upper-left block $\mathbf{A}$, will be Hermitian and positive-definite. Moreover, for each such matrix $\mathbf{X}$ it holds that $\|\mathbf{X}\| \leq 2\|\mathbf{M}_0\|$ and $\kappa(\mathbf{X}) \leq 2\kappa(\mathbf{M}_0)$. This value of $\mathbf{u}$ translates to

$$\log(1/\mathbf{u}) = O(\log(n)\log(\kappa))$$

required bits of precision.

Proof. At each recursive step, two matrices need to remain positive-definite: $\mathbf{A}$ and $\widetilde{\mathbf{S}}$. If $\mathbf{M}$ is positive-definite then so is $\mathbf{A}$. It remains to ensure the same for the Schur complement. To prove this we first fix $\mathbf{u}$ be bounded by the value

$$\mathbf{u} \leq \mathbf{u}_{++} := \frac{1}{n^{2c_{\mathsf{INV}}} \cdot \mathcal{E}_{\mathsf{INV}} \left(\frac{n}{2}, \mathbf{M}_0\right) \cdot 72\kappa(\mathbf{M}_0)^2}. \tag{13}$$

Assume that we are in the first level of recursion, i.e. $\mathbf{M} = \mathbf{M}_0$. Per Lemma C.2, we can write $\widetilde{\mathbf{S}} = \mathbf{S} + \mathbf{E}_{\mathbf{S}}$, where

$$\|\mathbf{E}_{\mathbf{S}}\| \leq \mathbf{u} \cdot \|\mathbf{M}_0\| \cdot \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \left(2\kappa(\mathbf{M}_0) + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{2}, \mathbf{M}_0\right)\right)^2 \left(3\kappa(\mathbf{M}_0) + \mathbf{u} \cdot \mathcal{E}_1 \left(\tfrac{n}{2}, \mathbf{M}_0\right)\right)\right).$$

For the assumed value of $\mathbf{u}$, both $\mathbf{u} \cdot \mathcal{E}_1 \left(\frac{n}{2}, \mathbf{M}_0\right) \leq 1$ and $\mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}} \left(\frac{n}{2}, \mathbf{M}_0\right) \leq 1$, and we also know that $1 \leq \kappa(\mathbf{M}_0)$, in which case the bound simplifies to

$$\begin{aligned}
\|\mathbf{E}_{\mathbf{S}}\| &\leq \mathbf{u} \cdot \|\mathbf{M}_0\| \cdot \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \left(2\kappa(\mathbf{M}_0) + (1+1)^2 \left(3\kappa(\mathbf{M}_0) + 1\right)\right) \\
&\leq \mathbf{u} \cdot \|\mathbf{M}_0\| \cdot \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \left(2\kappa(\mathbf{M}_0) + 4 \left(3\kappa(\mathbf{M}_0) + \kappa(\mathbf{M}_0)\right)\right) \\
&\leq \mathbf{u} \cdot \|\mathbf{M}_0\| \cdot \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot 18\kappa(\mathbf{M}_0) \\
&\leq \frac{1}{n^{2c_{\mathsf{INV}}} \cdot \mathcal{E}_{\mathsf{INV}} \left(\frac{n}{2}, \mathbf{M}_0\right) \cdot 72\kappa(\mathbf{M}_0)^2} \cdot \|\mathbf{M}_0\| \cdot \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot 18\kappa(\mathbf{M}_0) \\
&\leq \frac{1}{n^{2c_{\mathsf{INV}}}} \cdot \lambda_{\min}(\mathbf{M}_0). \tag{14}
\end{aligned}$$

Using this, we derive the following useful inequalities

$$\begin{aligned}
\lambda_{\max}(\widetilde{\mathbf{S}}) &\leq \|\mathbf{S}\| + \|\mathbf{E}_{\mathbf{S}}\| \leq (1 + 1/n^{2c_{\mathsf{INV}}})\|\mathbf{M}_0\|, \\
\lambda_{\min}(\widetilde{\mathbf{S}}) &\geq \lambda_{\min}(\mathbf{S}) - \|\mathbf{E}_{\mathbf{S}}\| \geq \lambda_{\min}(\mathbf{S}) - \frac{1}{n^{2c_{\mathsf{INV}}}}\lambda_{\min}(\mathbf{M}_0) \geq (1 - 1/n^{2c_{\mathsf{INV}}})\lambda_{\min}(\mathbf{M}_0), \\
\kappa(\widetilde{\mathbf{S}}) &\leq \frac{n^{2c_{\mathsf{INV}}} + 1}{n^{2c_{\mathsf{INV}}} - 1} \cdot \kappa(\mathbf{M}_0), \\
\mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{4}, \widetilde{\mathbf{S}}\right) &= \mu(n/4)\kappa(\widetilde{\mathbf{S}})^{c_{\mathsf{INV}} \log(n/4)} \\
&\leq \mu(n/4) \cdot \left(\frac{n^{2c_{\mathsf{INV}}} + 1}{n^{2c_{\mathsf{INV}}} - 1}\right)^{c_{\mathsf{INV}} \log(n/4)} \cdot \kappa(\mathbf{M}_0)^{c_{\mathsf{INV}} \log(n/4)} \\
&\leq n^{c_{\mathsf{INV}}} \cdot \mu(n/4) \cdot \kappa(\mathbf{M}_0)^{c_{\mathsf{INV}} \log(n/4)} \\
&\leq n^{c_{\mathsf{INV}}} \mathcal{E}_{\mathsf{INV}} \left(\tfrac{n}{4}, \mathbf{M}_0\right).
\end{aligned}$$

In the above, we used the fact that for $c_{\mathsf{INV}} \geq 1$ we have that $\frac{n^{2c_{\mathsf{INV}}}+1}{n^{2c_{\mathsf{INV}}}-1} \leq 2$, for all $n \geq 3$, and therefore $2^{c_{\mathsf{INV}} \log(n/4)} = (n/4)^{c_{\mathsf{INV}}} \leq n^{c_{\mathsf{INV}}}$. Since the smallest eigenvalue of $\widetilde{\mathbf{S}}$ is larger than a positive value, we can conclude that it is Hermitian and positive-definite, and its condition number and spectral norm are appropriately bounded.

In a similar manner, we can now get a bound for the Schur complement of the Schur complement. In this recursive step we call Algorithm 2 with $\mathbf{M} = \widetilde{\mathbf{S}}$. Let

$$\widetilde{\mathbf{S}} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1^* \\ \mathbf{B}_1 & \mathbf{C}_1 \end{pmatrix}.$$

Let $\mathbf{S}_1$ be the true Schur complement of $\widetilde{\mathbf{S}}$, i.e. $\mathbf{S}_1 = \mathbf{C}_1 - \mathbf{B}_1 \mathbf{A}_1^{-1} \mathbf{B}_1^*$, and let $\widetilde{\mathbf{S}}_1$ be the approximate Schur complement of $\widetilde{\mathbf{S}}$ that is constructed in line 7 of Algorithm 2 when executed on $\widetilde{\mathbf{S}}$. Let $\mathbf{E}_{\mathbf{S}_1}$ be the corresponding error matrix, i.e. $\widetilde{\mathbf{S}}_1 = \mathbf{S}_1 + \mathbf{E}_{\mathbf{S}_1}$. Using Lemma C.2 and simplifying the terms $(1 + 1/n^{c_{\text{INV}}}) \leq 2$ and $\frac{n^{2c_{\text{INV}}}+1}{n^{2c_{\text{INV}}}-1} \leq 2$ we have that

$$\|\mathbf{E}_{\mathbf{S}_1}\| \leq \mathbf{u} \ \overbrace{\|\widetilde{\mathbf{S}}\|}^{\leq 2\|\mathbf{M}_0\|} \ \underbrace{\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \widetilde{\mathbf{S}}\right)}_{\leq n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4},\mathbf{M}_0\right)} \left[2 \ \overbrace{\kappa(\widetilde{\mathbf{S}})}^{\leq 2\kappa(\mathbf{M}_0)} + \left(1 + \mathbf{u} \cdot \underbrace{\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \widetilde{\mathbf{S}}\right)}_{\leq n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4},\mathbf{M}_0\right)}\right)^2 \left(3 \ \overbrace{\kappa(\widetilde{\mathbf{S}})}^{\leq 2\kappa(\mathbf{M}_0)} + \mathbf{u} \ \overbrace{\mathcal{E}_1\left(\tfrac{n}{4}, \widetilde{\mathbf{S}}\right)}^{\leq 2\mathcal{E}_1\left(\tfrac{n}{2},\mathbf{M}_0\right)} \right) \right]$$

$$\leq \mathbf{u} \cdot 2\|\mathbf{M}_0\| \cdot n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \mathbf{M}_0\right) \left(4\kappa(\mathbf{M}_0) + \left(1 + \mathbf{u} \cdot n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \mathbf{M}_0\right)\right)^2 \left(6\kappa(\mathbf{M}_0) + \mathbf{u} \cdot 2\mathcal{E}_1\left(\tfrac{n}{4}, \mathbf{M}_0\right)\right) \right)$$

$$\leq \mathbf{u} \cdot 2\|\mathbf{M}_0\| \cdot n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \mathbf{M}_0\right) \left(4\kappa(\mathbf{M}_0) + (1+1)^2 \left(6\kappa(\mathbf{M}_0) + 2\right) \right)$$

$$\leq \mathbf{u} \cdot \|\mathbf{M}_0\| \cdot n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \mathbf{M}_0\right) \cdot 72\kappa(\mathbf{M}_0)$$

$$\leq \frac{1}{n^{2c_{\text{INV}}} \cdot \mathcal{E}_{\text{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot 72\kappa(\mathbf{M}_0)^2} \cdot \|\mathbf{M}_0\| \cdot n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{4}, \mathbf{M}_0\right) \cdot 72\kappa(\mathbf{M}_0)$$

$$= \frac{1}{n^{c_{\text{INV}}}} \cdot \lambda_{\min}(\mathbf{M}_0).$$

The corresponding bounds for $\widetilde{\mathbf{S}}_1$ can be derived:

$$\begin{aligned}
\lambda_{\max}(\widetilde{\mathbf{S}}_1) &\leq \|\widetilde{\mathbf{S}}\| + \|\mathbf{E}_{\mathbf{S}_1}\| \\
&\leq \|\mathbf{S}\| + \|\mathbf{E}_{\mathbf{S}}\| + \|\mathbf{E}_{\mathbf{S}_1}\| \\
&\leq \|\mathbf{M}_0\| + \frac{1}{n^{2c_{\text{INV}}}}\|\mathbf{M}_0\| + \frac{1}{n^{c_{\text{INV}}}}\|\mathbf{M}_0\| \\
&\leq (1 + 2/n^{c_{\text{INV}}})\|\mathbf{M}_0\|, \\
\lambda_{\min}(\widetilde{\mathbf{S}}_1) &\geq (1 - 2/n^{c_{\text{INV}}})\lambda_{\min}(\mathbf{M}_0), \\
\kappa(\widetilde{\mathbf{S}}_1) &\leq \frac{n^{c_{\text{INV}}} + 2}{n^{c_{\text{INV}}} - 2} \cdot \kappa(\mathbf{M}_0), \\
\mathcal{E}_{\text{INV}}\left(\tfrac{n}{8}, \widetilde{\mathbf{S}}_1\right) &= \mu(n/8)\kappa(\widetilde{\mathbf{S}})^{c_{\text{INV}} \log(n/8)} \\
&\leq \mu(n/8) \cdot \left(\frac{n^{c_{\text{INV}}} + 2}{n^{c_{\text{INV}}} - 2}\right)^{c_{\text{INV}} \log(n/8)} \cdot \kappa(\mathbf{M}_0)^{c_{\text{INV}} \log(n/8)} \\
&\leq n^{c_{\text{INV}}} \cdot \mu(n/8) \cdot \kappa(\mathbf{M}_0)^{c_{\text{INV}} \log(n/8)} \\
&= n^{c_{\text{INV}}}\mathcal{E}_{\text{INV}}\left(\tfrac{n}{8}, \mathbf{M}_0\right).
\end{aligned}$$

We can therefore identify that $\widetilde{\mathbf{S}}_1$ is also Hermitian positive-definite and its extremal eigenvalues are similarly bounded with those of $\widetilde{\mathbf{S}}$.

If we keep applying the same analysis for all $\log(n)$ iterations, as long as n is greater than some constant such that $(n^{c_{\text{INV}}} + \log n)/(n^{c_{\text{INV}}} - \log n) \leq 2$, then for each $i = 1, ..., \log n$ it holds that

$$\|\mathbf{E}_{\mathbf{S}_i}\| \leq \frac{1}{n^{c_{\text{INV}}}} \cdot \lambda_{\min}(\mathbf{M}_0) \leq \frac{1}{n^{c_{\text{INV}}}}\|\mathbf{M}_0\|,$$

which implies that

$$\lambda_{\max}(\widetilde{\mathbf{S}}_i) \leq (1 + i/n^{c_{\mathsf{INV}}})\|\mathbf{M}_0\|,$$

$$\lambda_{\min}(\widetilde{\mathbf{S}}_i) \geq (1 - i/n^{c_{\mathsf{INV}}})\lambda_{\min}(\mathbf{M}_0),$$

$$\kappa(\widetilde{\mathbf{S}}_i) \leq \frac{n^{c_{\mathsf{INV}}} + i}{n^{c_{\mathsf{INV}}} - i} \cdot \kappa(\mathbf{M}_0),$$

$$\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2^i}, \widetilde{\mathbf{S}}_i\right) \leq n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2^i}, \mathbf{M}_0\right).$$

The same bounds hold for the matrices $\mathbf{A}_i$, since they always originate from the top-left corner of a matrix $\widetilde{\mathbf{S}}_j$, where $j < i$, i.e., $\mathbf{A}_i$ either has no errors or it inherits the errors from a matrix $\widetilde{\mathbf{S}}_j$. We can therefore conclude that, for the value of $\mathbf{u}$ in Inequality (13), every matrix $\mathbf{A}_i$ and $\widetilde{\mathbf{S}}_i$ that is constructed during the recursion of Algorithm 2 will be Hermitian and positive-definite and its condition number and spectral norm will be at most $2\kappa(\mathbf{M}_0)$ and $2\|\mathbf{M}_0\|$ respectively. The required number of bits is

$$\log(1/\mathbf{u}_{++}) = \log\left(n^{2c_{\mathsf{INV}}} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot 72\kappa(\mathbf{M}_0)^2\right) = O\left(\log(n)\log(\kappa(\mathbf{M}_0))\right).$$

$\square$

C.3 Final backward-approximation bounds and proof of Theorem 1.2

Having safeguarded the possibility of a breakdown due to loss of positive-definiteness, we can now revisit the bounds of Lemma C.1 and finalize the proof of Theorem 1.2.

Theorem C.1 (Restatement of Theorem 1.2). *Given a Hermitian positive-definite matrix $\mathbf{M}$, there exists an algorithm $\mathbf{L} \leftarrow \mathsf{CHOLESKY}(\mathbf{M})$, listed in Algorithm 2, which requires $O(T_{\mathsf{MM}}(n))$ arithmetic operations. This algorithm is logarithmically stable, in a sense that, there exist global constants c_1, c_2, c_3, such that for all $\epsilon \in (0, 1)$, if executed in a floating point machine with precision*

$$\mathbf{u} \leq \mathbf{u}_{\mathsf{CHOLESKY}} := \epsilon\frac{1}{c_1 n^{c_2}\kappa(\mathbf{M})^{c_3\log n}},$$

which translates into $O\left(\log(n)\log(\kappa(\mathbf{M})) + \log(\tfrac{1}{\epsilon})\right)$ required bits of precision, then it does not break down due to arithmetic errors, and the solution returned satisfies $\|\mathbf{L}\mathbf{L}^* - \mathbf{M}\| \leq \epsilon\|\mathbf{M}\|$.

Proof. The arithmetic complexity of the algorithm was already bounded in Proposition C.2.

For the error proof we assume the bound for $\mathbf{u} \leq \mathbf{u}_{++}$ from Inequality (13). Combining the inequalities for a single recursion step from Lemma C.2 with the results of the previous section and the bound for $\mathbf{u}$, we have the following inequalities (recall once more that $\mathbf{M}$ is the input of the algorithm for the specific recursive call, while $\mathbf{M}_0$ is the matrix in the original call):

$$\left\|\mathbf{E}_3^{\mathsf{MM}}\right\| \leq \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2 \|\mathbf{L}_{11}\|$$

$$\leq \mathbf{u} \cdot 8 \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}_0\right)\|\mathbf{L}_{11}\|,$$

$$\|\mathbf{E}_{\mathbf{BA}i}\| \leq \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)$$

$$\leq \mathbf{u} \cdot \left(2n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \kappa(\mathbf{M}_0) + 4\mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}_0\right)\right)$$

$$\leq \mathbf{u} \cdot 6n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \kappa(\mathbf{M}_0),$$

$$\|\mathbf{E}_{\mathbf{L}_{21}}\| \leq \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left[\kappa(\mathbf{M}) + \left(2 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\right] \cdot \|\mathbf{L}_{11}\|$$

$$\leq \mathbf{u} \cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot (2\kappa(\mathbf{M}_0) + 9) \cdot \|\mathbf{L}_{11}\|$$

$$\leq \mathbf{u} \cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot 11\kappa(\mathbf{M}_0) \cdot \|\mathbf{L}_{11}\|,$$

$$\|\mathbf{E}_{\mathbf{S}}\| \leq \mathbf{u} \cdot \|\mathbf{M}\| \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right) \cdot \left(2\kappa(\mathbf{M}) + \left(1 + \mathbf{u} \cdot \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}\right)\right)^2\left(3\kappa(\mathbf{M}) + \mathbf{u} \cdot \mathcal{E}_1\left(\tfrac{n}{2}, \mathbf{M}\right)\right)\right)$$

$$\leq \mathbf{u} \cdot 2\|\mathbf{M}_0\| \cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \left(4\kappa(\mathbf{M}_0) + (1+1)^2\left(6\kappa(\mathbf{M}_0) + 1\right)\right)$$

$$\leq \mathbf{u} \cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot 64\kappa(\mathbf{M}_0) \cdot \|\mathbf{M}_0\|.$$

41

Using those inequalities, the first term that we need to bound from Lemma C.1 is the following

$$\|\mathbf{L}_{21}\mathbf{L}_{11}^* - \mathbf{B}\| \le (\kappa(\mathbf{M}) + \|\mathbf{E}_{\mathbf{BA}i}\|)\,\mathsf{err}(n/2) + \|\mathbf{E}_{\mathbf{BA}i}\|\|\mathbf{M}\| + \|\mathbf{E}_3^{\mathsf{MM}}\|\|\mathbf{L}_{11}^*\|$$

$$\le \left(2\kappa(\mathbf{M}_0) + \mathbf{u}\cdot 6n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\right)\mathsf{err}(n/2)$$

$$\ldots + \mathbf{u}\cdot 6n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\cdot 2\|\mathbf{M}_0\|$$

$$\ldots + \mathbf{u}\cdot 8\cdot\mathcal{E}_1\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot \underbrace{\|\mathbf{L}_{11}\|\|\mathbf{L}_{11}^*\|}_{\le\|\mathbf{A}\|+\|\mathbf{E}_{\mathbf{A}}^{\mathsf{CH}}\|\le 2\|\mathbf{M}_0\|+\mathsf{err}(n/2)}$$

$$\le \left(2\kappa(\mathbf{M}_0) + \mathbf{u}\cdot 6n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\right)\mathsf{err}(n/2)$$

$$\ldots + \mathbf{u}\cdot 12n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\|\mathbf{M}_0\|$$

$$\ldots + \mathbf{u}\cdot 16\cdot\mathcal{E}_1\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\|\mathbf{M}_0\|$$

$$\ldots + \mathbf{u}\cdot 8\cdot\mathcal{E}_1\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\mathsf{err}(n/2)$$

$$\le \left(2\kappa(\mathbf{M}_0) + \mathbf{u}\cdot 6n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0) + \mathbf{u}\cdot 8\cdot\mathcal{E}_1\left(\tfrac{n}{2},\mathbf{M}_0\right)\right)\mathsf{err}(n/2)$$

$$\ldots + \mathbf{u}\cdot\left(12n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0) + \mathbf{u}\cdot 16\cdot\mathcal{E}_1\left(\tfrac{n}{2},\mathbf{M}_0\right)\right)\cdot\|\mathbf{M}_0\|.$$

$$\le 22\cdot\kappa(\mathbf{M}_0)\cdot\mathsf{err}(n/2) + \mathbf{u}\cdot 28n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\cdot\|\mathbf{M}_0\|,$$

where in the last inequality we used the assumed bound for $\mathbf{u}$, namely Inequality (13).

Similarly

$$\|\mathbf{L}_{21}\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* - \mathbf{C}\| \le 2\kappa^2(\mathbf{M})\cdot\mathsf{err}(n/2) + 2\kappa(\mathbf{M})\|\mathbf{E}_{\mathbf{L}_{21}}\|\|\mathbf{L}_{11}\| + \|\mathbf{E}_{\mathbf{L}_{21}}\|^2 + \|\mathbf{E}_{\mathbf{S}}\|$$

$$\le 8\kappa^2(\mathbf{M}_0)\cdot\mathsf{err}(n/2)$$

$$\ldots + 4\kappa(\mathbf{M}_0)\cdot\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot 11\kappa(\mathbf{M}_0)\cdot\|\mathbf{L}_{11}\|\|\mathbf{L}_{11}\|$$

$$\ldots + \left(\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot 11\kappa(\mathbf{M}_0)\cdot\|\mathbf{L}_{11}\|\right)^2$$

$$\ldots + \mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot 64\kappa(\mathbf{M}_0)\cdot\|\mathbf{M}_0\|$$

$$\le 8\kappa^2(\mathbf{M}_0)\cdot\mathsf{err}(n/2)$$

$$\ldots + 44\kappa(\mathbf{M}_0)^2\cdot\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot(2\|\mathbf{M}_0\| + \mathsf{err}(n/2))$$

$$\ldots + \left(\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot 11\kappa(\mathbf{M}_0)\right)^2\cdot(2\|\mathbf{M}_0\| + \mathsf{err}(n/2))$$

$$\ldots + \mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot 64\kappa(\mathbf{M}_0)\cdot\|\mathbf{M}_0\|$$

$$\le \Bigg[8\kappa^2(\mathbf{M}_0)$$

$$\ldots + 44\kappa(\mathbf{M}_0)^2\cdot\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)$$

$$\ldots + 121\left(\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\right)^2\Bigg]\cdot\mathsf{err}(n/2)$$

$$\ldots + \Bigg[88\kappa(\mathbf{M}_0)^2\cdot\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)$$

$$\ldots + 242\left(\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\kappa(\mathbf{M}_0)\right)^2$$

$$\ldots + \mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot 64\kappa(\mathbf{M}_0)\Bigg]\cdot\|\mathbf{M}_0\|$$

$$= \Bigg[8 + \left(44 + 121\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\right)\cdot\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\Bigg]\cdot\kappa(\mathbf{M}_0)^2\cdot\mathsf{err}(n/2)$$

$$\ldots + \mathbf{u}\cdot\|\mathbf{M}_0\|\cdot\kappa(\mathbf{M}_0)\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\Bigg[88\kappa(\mathbf{M}_0) + 64 + \ldots$$

$$\ldots + 242\kappa(\mathbf{M}_0)\mathbf{u}\cdot n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\Bigg]$$

$$\le 173\cdot\kappa(\mathbf{M}_0)^2\cdot\mathsf{err}(n/2) + \mathbf{u}\cdot 394\kappa(\mathbf{M}_0)^2 n^{c_{\mathsf{INV}}}\mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2},\mathbf{M}_0\right)\cdot\|\mathbf{M}_0\|,$$

where in the last inequality we used again (13).

We can now derive a recursive formula for the main error bounds, Inequality (10).

$$
\begin{aligned}
\mathsf{err}(n) &= \|\mathbf{LL}^* - \mathbf{M}\| \\
&\leq \|\mathbf{L}_{11}\mathbf{L}_{11}^* - \mathbf{A}\| + \|\mathbf{L}_{21}\mathbf{L}_{11}^* - \mathbf{B}\| + \|\mathbf{L}_{21}\mathbf{L}_{21}^* + \mathbf{L}_{22}\mathbf{L}_{22}^* - \mathbf{C}\| \\
&\leq \mathsf{err}(n/2) + \dots \\
&\quad \dots + 22 \cdot \kappa(\mathbf{M}_0) \cdot \mathsf{err}(n/2) + \mathbf{u} \cdot 28 n^{c_{\mathsf{INV}}} \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \kappa(\mathbf{M}_0) \cdot \|\mathbf{M}_0\| + \dots \\
&\quad \dots + 173 \cdot \kappa(\mathbf{M}_0)^2 \cdot \mathsf{err}(n/2) + \mathbf{u} \cdot 394\kappa(\mathbf{M}_0)^2 n^{c_{\mathsf{INV}}} \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \|\mathbf{M}_0\| \\
&\leq 196\kappa(\mathbf{M}_0)^2 \cdot \mathsf{err}(n/2) + \mathbf{u} \cdot 422\kappa(\mathbf{M}_0)^2 n^{c_{\mathsf{INV}}} \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \|\mathbf{M}_0\|. \quad (15)
\end{aligned}
$$

The base case $n = 1$ is trivial since $\mathbf{M}$ is just a positive real number[2] and we can compute $\mathbf{L} = \mathbf{fl}(\sqrt{\mathbf{M}}) = (1 + \delta)\sqrt{\mathbf{M}}$ with $|\delta| \leq \mathbf{u}$, which means that $\mathbf{LL}^* = \mathbf{L}^2 = (1 + \delta)^2\mathbf{M} > 0$ and

$$
|\mathbf{M} - \mathbf{LL}^*| = |\mathbf{M}(2\delta + \delta^2)| \leq 3\mathbf{u}\|\mathbf{M}\| \leq 6\mathbf{u}\|\mathbf{M}_0\|.
$$

If we denote $\alpha = 196\kappa(\mathbf{M}_0)^2$ and $\beta = \mathbf{u} \cdot 422\kappa(\mathbf{M}_0)^2 n^{c_{\mathsf{INV}}} \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right)$, then the solution of the recursion can be written as

$$
\begin{aligned}
\mathsf{err}(n) &\leq \alpha^{\log n}\mathsf{err}(1) + \beta\|\mathbf{M}_0\| \sum_{i=0}^{\log n - 1} \alpha^i \\
&\leq \alpha^{\log n}6\mathbf{u}\|\mathbf{M}_0\| + \log(n)\alpha^{\log n}\beta\|\mathbf{M}_0\| \\
&= \mathbf{u}n^{c_1}\kappa(\mathbf{M}_0)^{2\log n}\|\mathbf{M}_0\| + \log(n)n^{c_1}\kappa(\mathbf{M}_0)^{2\log n}\mathbf{u} \cdot 422\kappa(\mathbf{M}_0)^2 n^{c_{\mathsf{INV}}} \mathcal{E}_{\mathsf{INV}}\left(\tfrac{n}{2}, \mathbf{M}_0\right) \cdot \|\mathbf{M}_0\| \\
&= \mathbf{u} \cdot c_4 n^{c_2}\kappa(\mathbf{M}_0)^{c_3 \log n} \cdot \|\mathbf{M}_0\|.
\end{aligned}
$$

We can now absorb the remaining terms inside $\mathbf{u}$, by tuning the constants of Inequality (13), to argue that we can achieve any desired (multiplicative) backward-accuracy $\epsilon \in (0, 1)$ and if we set

$$
\mathbf{u} \leq \epsilon \frac{1}{c_1' n^{c_2'} \kappa(\mathbf{M}_0)^{c_3' \log n}}
$$

for some constants c_1', c_2', c_3'. This translates to the advertised

$$
\log(1/\mathbf{u}) = O\left(\log(n^{c_2'}\kappa(\mathbf{M}_0)^{c_3' \log n}/\epsilon)\right) = O\left(\log(n)\log(\kappa(\mathbf{M}_0)) + \log(\tfrac{1}{\epsilon})\right)
$$

bits of precision. □

C.4 Reducing the GEP to a regular Hermitian eigenproblem

We can now use CHOLESKY to reduce the definite GEP to a regular Hermitian eigenproblem For the rest of the paper, we assume that $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\|, \|\mathbf{S}^{-1}\mathbf{H}\| \leq 1$. This is not a limitation since we can approximate the norms of $\mathbf{H}$ and $\mathbf{S}^{-1}$ in floating point using the algorithm SIGMAK, which is described later, and then scale accordingly. Formally, let $\eta \gtrsim \|\mathbf{H}\|$ and $\sigma \gtrsim \|\mathbf{S}^{-1}\|$. Then we can rewrite the generalized eigenproblem

$$
\mathbf{HC} = \mathbf{SC\Lambda} \quad \Leftrightarrow \quad (\tfrac{1}{\eta}\mathbf{H})\mathbf{C} = (\sigma\mathbf{S})\mathbf{C}(\mathbf{\Lambda}\tfrac{1}{\eta\sigma}) \quad \Leftrightarrow \quad \mathbf{H}'\mathbf{C} = \mathbf{S}'\mathbf{C\Lambda}',
$$

i.e. it is the same generalized eigenproblem only with scaled eigenvalues. We can thus safely make the unit-norms assumption. In Proposition C.3 we prove the properties of the reduction, which, in brief, states that we can compute in $O(T_{\mathsf{MM}}(n))$ floating point operations using $O(\log(n)\log(\kappa(\mathbf{S})) + \log(1/\epsilon))$ bits the matrix $\widetilde{\mathbf{H}} \leftarrow \widetilde{\mathbf{L}}^*\mathbf{H}\widetilde{\mathbf{L}}$, where $\widetilde{\mathbf{L}}$ is returned by CHOLESKY(INV($\mathbf{S}$)). The eigenvalues of $\widetilde{\mathbf{H}}$ provably approximate the pencil eigenvalues:

$$
\left|\lambda_i(\widetilde{\mathbf{H}}) - \lambda_i(\mathbf{H}, \mathbf{S})\right| \leq \epsilon, \quad (16)
$$

where $\epsilon \in (0, 1)$ is a given accuracy. Algorithm 4 details the reduction.

---

[2]It has to be real otherwise a complex 1-by-1 matrix is not Hermitian.

Algorithm 4: REDUCE.

Proposition C.3. *Given $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}^n_{++}$, and $\epsilon \in (0, 1)$, Algorithm 4 executes $O(T_{\mathsf{MM}}(n))$ floating point operations and returns a matrix $\widetilde{\mathbf{H}} \leftarrow \text{REDUCE}(\mathbf{H}, \mathbf{S}, \epsilon)$, such that, if the machine precision satisfies*

$$\mathbf{u} \leq \mathbf{u}_{\mathsf{REDUCE}} := \epsilon \frac{1}{\rho_1 n^{\rho_2} (4\kappa(\mathbf{S}))^{\rho_3 \log(n)}},$$

for some constants ρ_1, ρ_2, ρ_3, and $\epsilon \in (0, 1)$, which translates to

$$O\left(\log(1/\mathbf{u}_{\mathsf{REDUCE}})\right) = O\left(\log(n)\log(\kappa(\mathbf{S})) + \log(1/\epsilon)\right)$$

required bits of precision, then for all $i \in [n]$

$$|\lambda_i(\widetilde{\mathbf{H}}) - \lambda_i(\mathbf{H}, \mathbf{S})| \leq \epsilon \|\mathbf{S}^{-1}\| \|\mathbf{H}\|,$$

where $\lambda_i(\mathbf{H}, \mathbf{S})$ are the eigenvalues of the Hermitian definite pencil $(\mathbf{H}, \mathbf{S})$.

Proof. Let $\mathbf{L}$ be the (exact) lower triangular Cholesky factor of $\mathbf{S}^{-1}$, i.e. $\mathbf{L}\mathbf{L}^* = \mathbf{S}^{-1}$ (which is unique up to column phases). Expanding the equation in line 3 of Algorithm 4 we have

$$\begin{aligned}
\Lambda(\widetilde{\mathbf{H}}) &= \Lambda\left(\widetilde{\mathbf{L}}^*\mathbf{H}\widetilde{\mathbf{L}} + \mathbf{E}_5\right) \\
&= \Lambda\left(\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*\mathbf{H} + \widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1}\right) \\
&= \Lambda\left((\mathbf{S}^{-1} + \mathbf{E}_1^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}})\mathbf{H} + \widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1}\right) \\
&= \Lambda\left(\mathbf{S}^{-1}\mathbf{H} + \mathbf{E}_1^{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{CH}}\mathbf{H} + \widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1}\right) \\
&= \Lambda\left(\mathbf{L}^*\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\mathbf{E}_1^{\mathsf{INV}}\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\mathbf{E}_2^{\mathsf{CH}}\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1}\mathbf{L}\right).
\end{aligned}$$

At the same time, $\Lambda(\mathbf{S}^{-1}\mathbf{H}) = \Lambda(\mathbf{L}^*\mathbf{H}\mathbf{L})$. Combining this with Kahan's bound from Fact A.1 and denoting

$$\mathbf{B} = \mathbf{L}^{-1}\mathbf{E}_1^{\mathsf{INV}}\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\mathbf{E}_2^{\mathsf{CH}}\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1}\mathbf{L},$$

we have that

$$\left|\lambda_i(\mathbf{H}, \mathbf{S}) - \lambda_i(\widetilde{\mathbf{H}})\right| = |\lambda_i(\mathbf{L}^*\mathbf{H}\mathbf{L}) - \lambda_i(\mathbf{L}^*\mathbf{H}\mathbf{L} + \mathbf{B})| \leq C\|\mathbf{B}\|\log(n), \qquad (17)$$

for some constant C. It remains to bound for $\|\mathbf{B}\|$.

(i) $\mathbf{E}_1^{\mathsf{INV}}$ originates from the inversion of $\mathbf{S}_{\mathsf{INV}}$. Let $\mathbf{S}_{\mathsf{INV}} = \mathsf{HERM}(\mathsf{INV}(\mathbf{S})) = \mathbf{S}^{-1} + \mathbf{E}_1^{\mathsf{INV}}$. From Corollary A.3, we know that as long as the machine precision satisfies

$$\mathbf{u}_{\mathsf{REDUCE}} \leq \delta \frac{1}{c_{\mathsf{HERM}}\log(n)\mu_{\mathsf{INV}}(n)\kappa(\mathbf{S})^{c_{\mathsf{INV}}\log(n)+1}},$$

for some $\delta \in (0, 1/2)$, then all of the following hold: $\|\mathbf{E}_1^{\mathsf{INV}}\| \leq \delta\|\mathbf{S}^{-1}\|$, $\frac{1}{2}\|\mathbf{S}^{-1}\| \leq \|\mathbf{S}_{\mathsf{INV}}\| \leq 2\|\mathbf{S}^{-1}\|$, $\frac{1}{4}\kappa(\mathbf{S}) \leq \kappa(\mathbf{S}_{\mathsf{INV}}) \leq 4\kappa(\mathbf{S})$.

(ii) As a second step, the Cholesky factor $\widetilde{\mathbf{L}} = \mathsf{CHOLESKY}(\mathbf{S}_{\mathsf{INV}})$ is computed, such that $\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^* = \mathbf{S}_{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}$. From Theorem 1.2 we know that as long as

$$\mathbf{u}_{\mathsf{REDUCE}} \leq \delta \frac{1}{c_1 n^{c_2}\kappa(\mathbf{S}_{\mathsf{INV}})^{c_3\log(n)}},$$

for some constants c_1, c_2, c_3, then $\|\mathbf{E}_2^{\mathsf{CH}}\| \leq \delta\|\mathbf{S}_{\mathsf{INV}}\| \leq 2\delta\|\mathbf{S}^{-1}\|$. From the bound of $\kappa(\mathbf{S}_{\mathsf{INV}})$ in (i), the following is sufficient:

$$\mathbf{u}_{\mathsf{REDUCE}} \leq \delta \frac{1}{c_1 n^{c_2}\left(4\kappa(\mathbf{S})\right)^{c_3\log(n)}},$$

(iii) Finally, we form the matrix $\widetilde{\mathbf{H}} = \mathsf{HERM}\left(\mathsf{MM}(\mathsf{MM}(\widetilde{\mathbf{L}}, \mathbf{H}), \widetilde{\mathbf{L}}^*)\right) = \widetilde{\mathbf{L}}^*\mathbf{H}\widetilde{\mathbf{L}} + \mathbf{E}_3^{\mathsf{MM}}\widetilde{\mathbf{L}} + \mathbf{E}_4^{\mathsf{MM}} = \mathbf{E}_3$. Using Theorem A.2 and Corollary A.2, as long as

$$\mathbf{u}_{\mathsf{REDUCE}} \leq \delta \frac{1}{\mu(n)} = \delta \frac{1}{c_{\mathsf{HERM}}\log(n)\mu_{\mathsf{MM}}(n)},$$

$$\|\mathbf{E}_5\| = \|\widetilde{\mathbf{H}} - \widetilde{\mathbf{L}}^*\mathbf{H}\widetilde{\mathbf{L}}\| = \left\|\widetilde{\mathbf{L}}^*\mathbf{H}\widetilde{\mathbf{L}} + \mathbf{E}_3^{\mathsf{MM}}\widetilde{\mathbf{L}} + \mathbf{E}_4^{\mathsf{MM}} - \widetilde{\mathbf{L}}^*\mathbf{H}\widetilde{\mathbf{L}}\right\|$$

$$= \left\|\mathbf{E}_3^{\mathsf{MM}}\widetilde{\mathbf{L}} + \mathbf{E}_4^{\mathsf{MM}}\right\|$$

$$\leq \overbrace{\|\mathbf{E}_3^{\mathsf{MM}}\|}^{\leq \mathbf{u}\cdot\mu_{\mathsf{MM}}(n)\|\widetilde{\mathbf{L}}^*\|\|\mathbf{H}\|}\|\widetilde{\mathbf{L}}\| + \overbrace{\|\mathbf{E}_4^{\mathsf{MM}}\|}^{\leq c_{\mathsf{HERM}}\log(n)\cdot\mathbf{u}\mu_{\mathsf{MM}}(n)\|\widetilde{\mathbf{L}}^*+\mathbf{E}_3^{\mathsf{MM}}\|\cdot\|\widetilde{\mathbf{L}}\|}$$

$$\leq \mathbf{u}\cdot\mu_{\mathsf{MM}}(n)\|\widetilde{\mathbf{L}}\|^2\|\mathbf{H}\| + c_{\mathsf{HERM}}\log(n)\cdot\mathbf{u}\mu_{\mathsf{MM}}(n)\|\widetilde{\mathbf{L}}^*\mathbf{H} + \mathbf{E}_3^{\mathsf{MM}}\|\cdot\|\widetilde{\mathbf{L}}\|$$

$$\leq \mathbf{u}\cdot\mu(n)\|\widetilde{\mathbf{L}}\|^2\|\mathbf{H}\| + \mathbf{u}\mu(n)\|\widetilde{\mathbf{L}}^*\mathbf{H}\|\|\widetilde{\mathbf{L}}\| + \mathbf{u}\mu(n)\|\mathbf{E}_3^{\mathsf{MM}}\|\cdot\|\widetilde{\mathbf{L}}\|$$

$$\leq 2\mathbf{u}\cdot\mu(n)\|\widetilde{\mathbf{L}}\|^2\|\mathbf{H}\| + \mathbf{u}\mu(n)\cdot\mathbf{u}\cdot\mu_{\mathsf{MM}}(n)\|\widetilde{\mathbf{L}}^*\|\|\mathbf{H}\|\cdot\|\widetilde{\mathbf{L}}\|$$

$$= \mathbf{u}\cdot\mu(n)\|\widetilde{\mathbf{L}}\|^2\|\mathbf{H}\|\left(2 + \mathbf{u}\cdot\mu_{\mathsf{MM}}(n)\right)$$

$$= 3\mathbf{u}\cdot\mu(n)\overbrace{\|\widetilde{\mathbf{L}}\|^2}^{\leq \|\mathbf{S}_{\mathsf{INV}}\|+\|\mathbf{E}_2^{\mathsf{CH}}\|}\|\mathbf{H}\|$$

$$\leq 3\mathbf{u}\cdot\mu(n)\left(\|\mathbf{S}^{-1}\| + \|\mathbf{E}_1^{\mathsf{INV}}\| + \|\mathbf{E}_2^{\mathsf{CH}}\|\right)\|\mathbf{H}\|$$

$$\leq 3\mathbf{u}\cdot\mu(n)\left(2\|\mathbf{S}^{-1}\| + 2\delta\|\mathbf{S}^{-1}\|\right)\|\mathbf{H}\|$$

$$\leq 12\mathbf{u}\mu(n)\|\mathbf{S}^{-1}\|\|\mathbf{H}\|$$

$$\leq 12\delta\|\mathbf{S}^{-1}\|\|\mathbf{H}\|.$$

(iv) For the final desired error we need to bound $\|\widetilde{\mathbf{L}}\|\|\widetilde{\mathbf{L}}^{-1}\|$. Taking the square:

$$(\|\widetilde{\mathbf{L}}\|\|\widetilde{\mathbf{L}}^{-1}\|)^2 = \|\widetilde{\mathbf{L}}\|^2\|\widetilde{\mathbf{L}}^{-1}\|^2 = \|\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*\|\|(\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*)^{-1}\| = \|\mathbf{S}^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}\|\|(\mathbf{S}^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}})^{-1}\|$$

$$= \kappa(\mathbf{S}^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}).$$

For the spectral norm we have that $\|\mathbf{S}_{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}\| \leq \|\mathbf{S}_{\mathsf{INV}}\| + \|\mathbf{E}_2^{\mathsf{CH}}\| \leq 2\|\mathbf{S}^{-1}\| + 2\delta\|\mathbf{S}^{-1}\| \leq 3\|\mathbf{S}^{-1}\|$. For the spectral norm of the inverse we need to bound the smallest singular value of

$\mathbf{S}^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}$ from below:

$$\sigma_{\min}(\mathbf{S}^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}) \geq \sigma_{\min}(\mathbf{S}^{\mathsf{INV}}) - \|\mathbf{E}_2^{\mathsf{CH}}\| \geq \tfrac{1}{2}\sigma_{\min}(\mathbf{S}^{-1}) - 2\delta\|\mathbf{S}^{-1}\| = \tfrac{1}{2\|\mathbf{S}\|} - 2\delta\|\mathbf{S}^{-1}\|$$
$$= \tfrac{1 - 4\delta\kappa(\mathbf{S})}{2\|\mathbf{S}\|}.$$

Combining the last two bounds, that we have

$$\kappa(\mathbf{S}^{\mathsf{INV}} + \mathbf{E}_2^{\mathsf{CH}}) \leq \frac{3\|\mathbf{S}^{-1}\|}{\frac{1 - 4\delta\kappa(\mathbf{S})}{2\|\mathbf{S}\|}} = \frac{6\kappa(\mathbf{S})}{1 - 4\delta\kappa(\mathbf{S})}.$$

We now have bounds for all the required quantities to bound $\|\mathbf{B}\|$:

$$\|\mathbf{B}\| = \left\| \mathbf{L}^{-1}\mathbf{E}_1^{\mathsf{INV}}\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\mathbf{E}_2^{\mathsf{CH}}\mathbf{H}\mathbf{L} + \mathbf{L}^{-1}\widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1}\mathbf{L} \right\|$$

$$\leq \sqrt{\kappa(\mathbf{S})} \left\| \mathbf{E}_1^{\mathsf{INV}}\mathbf{H} + \mathbf{E}_2^{\mathsf{CH}}\mathbf{H} + \widetilde{\mathbf{L}}\mathbf{E}_5\widetilde{\mathbf{L}}^{-1} \right\|$$

$$\leq \sqrt{\kappa(\mathbf{S})} \left(\|\mathbf{E}_1^{\mathsf{INV}}\|\|\mathbf{H}\| + \|\mathbf{E}_2^{\mathsf{CH}}\mathbf{H}\| + \|\mathbf{E}_5\|\|\widetilde{\mathbf{L}}\|\|\widetilde{\mathbf{L}}^{-1}\| \right)$$

$$\leq \sqrt{\kappa(\mathbf{S})} \left(\delta\|\mathbf{S}^{-1}\|\|\mathbf{H}\| + 2\delta\|\mathbf{S}^{-1}\|\|\mathbf{H}\| + 12\delta\|\mathbf{S}^{-1}\|\|\mathbf{H}\|\sqrt{\frac{6\kappa(\mathbf{S})}{1 - 4\delta\kappa(\mathbf{S})}} \right)$$

$$\leq \delta\sqrt{\kappa(\mathbf{S})}\|\mathbf{S}^{-1}\|\|\mathbf{H}\| \left(3 + 12\sqrt{\frac{6\kappa(\mathbf{S})}{1 - 4\delta\kappa(\mathbf{S})}} \right).$$

Setting $\delta = \epsilon\frac{1}{64C\log(n)\kappa(\mathbf{S})}$, where C is the constant from Inequality (17) and $\epsilon \in (0, 1)$, this finally gives

$$\|\mathbf{B}\| \leq \epsilon\frac{1}{64C\log(n)\kappa(\mathbf{S})} \cdot \sqrt{\kappa(\mathbf{S})}\|\mathbf{S}^{-1}\|\|\mathbf{H}\| \left(3 + 12\sqrt{\frac{6\kappa(\mathbf{S})}{1 - 4\epsilon\frac{1}{64C\log(n)\kappa(\mathbf{S})}\kappa(\mathbf{S})}} \right)$$

$$= \epsilon\frac{1}{64C\log(n)\sqrt{\kappa(\mathbf{S})}}\|\mathbf{S}^{-1}\|\|\mathbf{H}\| \left(3 + 12\sqrt{\frac{6\kappa(\mathbf{S})}{1 - \epsilon\frac{1}{16}}} \right)$$

$$\leq \epsilon\frac{1}{64C\log(n)\sqrt{\kappa(\mathbf{S})}}\|\mathbf{S}^{-1}\|\|\mathbf{H}\| \left(3 + 12\sqrt{\frac{96}{15}\kappa(\mathbf{S})} \right)$$

$$\leq \epsilon\frac{1}{64C\log(n)\sqrt{\kappa(\mathbf{S})}}\|\mathbf{S}^{-1}\|\|\mathbf{H}\|(3 + 36\sqrt{\kappa(\mathbf{S})})$$

$$\leq \epsilon\frac{1}{C\log(n)}\|\mathbf{S}^{-1}\|\|\mathbf{H}\|,$$

which we can use in Inequality (17) to obtain

$$|\lambda_i(\mathbf{H}, \mathbf{S}) - \lambda_i(\widetilde{\mathbf{H}})| \leq \epsilon\|\mathbf{S}^{-1}\|\|\mathbf{H}\|.$$

We can now gather all the requirements for $\mathbf{u}_{\mathsf{REDUCE}}$. From the above, $\mathbf{u}_{\mathsf{REDUCE}}$ needs to satisfy

$$\mathbf{u}_{\mathsf{REDUCE}} \leq \epsilon\frac{1}{64C\log(n)\kappa(\mathbf{S})} \min \left\{ \frac{1}{c_{\mathsf{HERM}}\log(n)\mu_{\mathsf{INV}}(n)\kappa(\mathbf{S})^{c_{\mathsf{INV}}\log(n)+1}}, \right.$$
$$\left. \frac{1}{c_1 n^{c_2}(4\kappa(\mathbf{S}))^{c_3\log(n)}}, \frac{1}{c_{\mathsf{HERM}}\log(n)\mu_{\mathsf{MM}}(n)} \right\}$$

$$\leq \epsilon\frac{1}{\rho_1 n^{\rho_2}(4\kappa(\mathbf{S}))^{\rho_3\log(n)}},$$

for some suitably chosen constants ρ_1, ρ_2, ρ_3. This translates to

$$O\left(\log(1/\mathbf{u}_{\mathsf{REDUCE}})\right) = O\left(\log(n)\log(\kappa(\mathbf{S})) + \log(1/\epsilon)\right)$$

required bits of precision. Having established the eigenvalue bounds, note that $\lambda_i(\mathbf{H}, \mathbf{S}) = \lambda_i(\mathbf{L}^*\mathbf{H}\mathbf{L}, \mathbf{I})$. Since $\widetilde{\mathbf{H}}$ and $\mathbf{L}^*\mathbf{H}\mathbf{L}$ are both Hermitian, the eigenvalue bound holds also for the singular values. For scaled matrices $\|\mathbf{S}^{-1}\|, \|\mathbf{H}\| \leq 1$ we obtain a maximum ϵ additive error for the eigenvalues. $\qquad\square$

C.5 Counterexample for floating point LU

In this section provide a counter-example which illustrates why the backward-stable LU factorization algorithm of [39] is not sufficient to obtain a Cholesky factor for a Hermitian postive-definite matrix. We apply one-by-one the steps of the LUR algorithm of [39] on the matrix:

$$\mathbf{A} = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}.$$

(a) In the first step the matrices $\mathbf{L}_L$ and $\mathbf{U}_L$ are computed by calling $\mathsf{LUR}(\mathbf{A}_{1:2,1})$, to compute the LU factorization of the first column of $\mathbf{A}$. It is easy to see that this returns $\mathbf{L}_L = \begin{pmatrix} 1 \\ \frac{1}{3}(1 + \theta_1) \end{pmatrix}$ where $|\theta_1| \leq \mathbf{u}$, and $\mathbf{U}_L = 3$ (the $\mathbf{U}_L$ factor is a 1-by-1 matrix in this case).

(b) The second step updates the upper right corner of $\mathbf{A}$, by multiplying from the left with the inverse of the top-left block of $\mathbf{L}_L$. Assuming that multiplication and inversion with the number 1 does not incur errors, then for this specific example $\mathbf{A}_{1,2}$ is not modified in this step, i.e., it remains equal to 1, without errors.

(c) The third step updates the Schur complement, which becomes $\mathbf{A}_{2,2} \leftarrow \mathbf{fl}(\mathbf{A}_{2,2} - \frac{1}{3}(1 + \theta_1)) = (3 - \frac{1}{3}(1 + \theta_1))(1 + \theta_2)$, where $|\theta_2| \leq \mathbf{u}$ is the error term incurred by subtraction.

(d) The next step computes the $\mathbf{L}_R, \mathbf{U}_R \leftarrow \mathsf{LUR}(\mathbf{A}_{2,2})$ of the (updated) Schur complement. In this specific case it trivially returns $\mathbf{L}_R = 1$ and $\mathbf{U}_R = \mathbf{A}_{2,2} = (3 - \frac{1}{3}(1 + \theta_1))(1 + \theta_2)$.

(e) The final steps combine together the left and right LU factors to finally return $\mathbf{L} = \begin{pmatrix} 1 & 0 \\ \frac{1}{3}(1 + \theta_1) & 1 \end{pmatrix}$,

(f) and $\mathbf{U} = \begin{pmatrix} 3 & 1 \\ 0 & (3 - \frac{1}{3}(1 + \theta_1))(1 + \theta_2) \end{pmatrix}$.

Then, in exact arithmetic, $\mathbf{LU} = \begin{pmatrix} 3 & 1 \\ 1 + \theta_1 & 3(1 + \theta_2) + \theta_2(1 + \theta_1)\frac{1}{3} \end{pmatrix}$, which is not symmetric.

D   Regularization with diagonal disorder

We now arrive to the more interesting part of the analysis. In Section 2 we argued that if we know the gap and the midpoint then the sign function can yield the desired spectral projector. To compute the gap and midpoint efficiently, we will take advantage of the symmetry induced by the Cholesky-based reduction, and use the Wegner estimate [138] for Hermitian diagonal perturbations to regularize the problem in the spirit of smoothed analysis. We use the following variant of the Wegner estimate.

Proposition D.1. *Let $\mathbf{G}$ be a random diagonal matrix with independent diagonal elements sampled from $\mathcal{N}(0, \sigma^2)$, for $\sigma = \frac{1}{2\sqrt{2\ln(4n/\delta)}}$ and some $\delta \in (0, 1/2)$. Then for any interval $I \subset \mathbb{R}$*

$$\Pr\left[|\Lambda(\mathbf{H} + \mathbf{G}) \cap I| \geq 1\right] \leq \sqrt{4\pi \ln(4n/\delta)}n|I|,$$

where $\mathbf{H}$ is a fixed Hermitian matrix and $I \subset \mathbb{R}$ is a fixed interval.

Proof. This directly comes from the well-known result of Wegner [138], which states that that for any interval $I \subset \mathbb{R}$

$$\Pr\left[|\Lambda(\mathbf{H} + \mathbf{G})| \cap I| \geq 1\right] \leq \pi\|\rho\|_\infty n|I| \leq \sqrt{\frac{\pi}{2\sigma^2}}n|I|,$$

where $\|\rho\|_\infty = \frac{1}{\sqrt{2\pi\sigma^2}}$ is the supremum the probability density. $\qquad\square$

Similar results can be obtained for other classes of random matrices $\mathbf{G}$. Such an example is the Gaussian Unitary Ensemble (GUE) [4], in which case $\mathbf{G}$ is dense, but it is invariant under rotations which might be useful for other applications.

D.1 Sampling Gaussians and perturbing in floating point

The next step is to describe how to use such random diagonal perturbations in floating point. Briefly, we first assume that we have a fixed number of bits for the floating point exponent and for the mantissa. If we sample a standard normal Gaussian (in infinite precision), it can happen that it is too large to fit in the given number of bits. This is accounted for in the failure probability, which is very small due to the decay of the normal distribution. We describe our sampling method in the following definition, which can be thought of as sampling only the most significant bits of the floating point representation of a Gaussian.

Definition D.1. *A floating point standard normal sampler* $\mathsf{N}(p, t)$ *takes as input the number of exponent bits p, and the number of mantissa bits t, and returns a floating point number $\widetilde{g} \leftarrow \mathbf{fl}(g)$ where g is sampled from $\mathcal{N}(0, 1)$. Following the definitions of the floating point model in Appendix A.2, if $|g| \in [2^{-M}, 2^M(2 - 2^{-t})]$, where $M = 2^{p-1}$, then $|\widetilde{g} - g| \leq 2^{-t}|g| = \mathbf{u}|g|$. If $|g| > 2^M(2 - 2^{-t})$ the sampler returns $\widetilde{g} = \pm\mathsf{INF}$, i.e. the floating point representation of infinity, and if $|g| < 2^{-M}$ it returns zero.*

Lemma D.1 (Diagonal Gaussian sampler). *Let $\gamma \in (0, 1/2)$ and $\sigma > 0$, and assume that we want to obtain a floating representation of the matrix $\gamma\mathbf{G}$, where $\mathbf{G}$ is a diagonal matrix with independent diagonal entries from $\mathcal{N}(0, \sigma^2)$. Let $\mathsf{N}(p, t)$ be a floating point standard normal sampler as in Definition D.1, and $\mathbf{u} = 2^{-t}$. Let $\widetilde{\sigma}$ be such that $|\widetilde{\sigma} - \sigma| \leq \eta_c \sigma$, where $|\eta_c| \leq \frac{c\mathbf{u}}{1 - c\mathbf{u}}$ for some (integer) constant $c > 1$, and*

$$\widetilde{\mathbf{G}} = \mathrm{diag}(\widetilde{g}_1, \widetilde{g}_2, \ldots, \widetilde{g}_n),$$

where $\widetilde{g}_i = \mathsf{N}(p, t)$. If $p \geq C\log(\log(\frac{n}{\delta}))$ for some global constant C, then the diagonal matrix $\widetilde{\mathbf{V}} = \mathbf{fl}(\gamma \cdot \widetilde{\sigma} \cdot \widetilde{\mathbf{G}}) = \gamma\mathbf{G} + \mathbf{E}$, has all the following properties with probability at least $1 - \delta$:

(i) $\widetilde{\mathbf{V}}_{i,i} \notin \{0, \mathsf{INF}\}$,

(ii) $\mathbf{E}$ *is diagonal and* $\|\mathbf{E}\| \leq \gamma\eta_{c+2}\sigma\sqrt{2\ln(4n/\delta)}$,

(iii) $\|\widetilde{\mathbf{V}}\| \leq \gamma(1 + \eta_{c+2})\sigma\sqrt{2\ln(4n/\delta)}$.

Proof. Let us first analyze the conditions that are necessary such that all the sampled numbers lie within the floating point bounds from Appendix A.2, specifically, $|g_i| \in [2^{-M}, 2^M(2 - 2^{-t})]$, $i \in [n]$, where $M = 2^{p-1}$. For simplicity we consider $|g_i| \in [2^{-M}, 2^M]$.

For the lower bound, we have that

$$\Pr[|g_i| \leq 2^{-M}] \leq 2\frac{1}{\sqrt{2\pi}}2^{-M},$$

where we naively upper bounded the standard normal probability density in the interval $[0, 2^{-M}]$ by the constant function $\frac{1}{\sqrt{2\pi}}$. For $M = \log_2\left(\frac{2n}{\delta}\sqrt{\frac{2}{\pi}}\right)$ this implies that there are no subnormal g_i for all $i \in [n]$ simultaneously. This value of M translates to $p = O\left(\log(\log(\frac{n}{\delta}))\right)$ bits.

For the upper bound, a standard normal tail bound gives

$$\Pr[|g_i| \geq 2^M] \leq 2\exp(-2^{2M}/2). \tag{18}$$

For the aforementioned value of M, the probability that each one of the g_i is larger than 2^M is exponentially small in δ/n, therefore we can conclude that the sampler does not return any INF values with exponentially high probability.

This is already enough to argue that the sampler returns floating point numbers that are not subnormal and not INF. However, we will need a tighter bound for the magnitude of the g_i to bound the norm of the diagonal random matrix. Analyzing Eq (18) for $M = \frac{1}{2}\log_2(2\ln(\frac{4n}{\delta}))$, and taking a union bound over all $i \in [n]$ we have that $|g_i| \leq \sqrt{2\ln(\frac{4n}{\delta})} \ll 2^M$ holds for all i simultaneously with probability at least $1 - \delta/2$.

Conditioning on the event that all the g_i are in the correct range, from Definition D.1, $|\widetilde{g}_i - g_i| \leq \mathbf{u}|g_i|$ holds for all i. Then

$$\widetilde{v}_i = \mathbf{fl}(\widetilde{\sigma} \cdot \widetilde{g}_i) = (1 + \theta)\widetilde{\sigma}\widetilde{g}_i = \sigma g_i(1 + \eta_{c+1}),$$

48

where $c > 1$ is a constant and $|\eta_{c+1}| \leq \frac{(c+1)\mathbf{u}}{1-(c+1)\mathbf{u}}$. This means that every diagonal element of $\mathbf{fl}(\widetilde{\sigma}\widetilde{\mathbf{G}})$ is $(1 + \eta_{c+1})$-far from a random variable that is sampled from $\mathcal{N}(0, \sigma^2)$. Multiplying each diagonal element of $\mathbf{fl}(\widetilde{\sigma}\widetilde{\mathbf{G}})$ with γ to form $\widetilde{\mathbf{V}}$ (similar to the Step 4 of Algorithm 5) simply increases the (relative) error to $(1 + \eta_{c+2})$. Then we can write

$$\widetilde{\mathbf{V}} = \gamma \mathbf{V} + \mathbf{E},$$

where $\mathbf{V}$ is diagonal with independent diagonal entries from $\mathcal{N}(0, \sigma^2)$ and $\mathbf{E}$ is a diagonal matrix with diagonal elements $\mathbf{E}_{i,i}$ bounded in magnitude by

$$|\mathbf{E}_{i,i}| \leq \frac{(c+2)\mathbf{u}}{1-(c+2)\mathbf{u}}\gamma|\mathbf{V}_{i,i}|.$$

It also holds that $\|\mathbf{V}\| \leq \sigma\sqrt{2\ln(4n/\delta)}$ since the diagonal elements $\mathbf{V}_{i,i}$ since they are just the g_i's scaled by σ, and we already bounded $|g_i| \leq \sqrt{2\ln(4n/\delta)}$, which implies that $\|\mathbf{E}\| \leq \gamma\frac{(c+2)\mathbf{u}}{1-(c+2)\mathbf{u}}\sigma\sqrt{2\ln(4n/\delta)}$. Finally

$$\|\widetilde{\mathbf{V}}\| \leq \gamma\|\mathbf{V}\| + \|\mathbf{E}\| \leq \gamma\left(1 + \frac{(c+2)\mathbf{u}}{1-(c+2)\mathbf{u}}\right)\sigma\sqrt{2\ln(4n/\delta)}.$$

We conditioned only on two random events, where each holds with probability at least $1 - \delta/2$, giving the final success probability of at least $1 - \delta$. $\qquad\square$

REGULARIZE

Input: Hermitian matrix $\mathbf{A} \in \mathbb{H}^n$, perturbation scale factor $\gamma \in (0, 1/4)$, failure probability parameter $\delta \in (0, 1/4)$.

Requires: $\|\mathbf{A}\| \leq 1$.

Algorithm: $\widetilde{\mathbf{X}} \leftarrow$ REGULARIZE$(\mathbf{A}, \gamma, \delta)$.

1: $\widetilde{\sigma} = \mathbf{fl}\left(\frac{1}{2\sqrt{2\ln(4n/\delta)}}\right)$.

2: $p \leftarrow \lceil C\log(\log(\frac{n}{\delta}))\rceil$.

3: $\widetilde{\mathbf{G}} = \mathrm{diag}(\widetilde{g}_1, \ldots, \widetilde{g}_n), \widetilde{g}_i \leftarrow \mathsf{N}(p, \log(1/\mathbf{u}))$

4: $\widetilde{\mathbf{V}} = \mathbf{fl}(\gamma \cdot \widetilde{\sigma} \cdot \widetilde{\mathbf{G}}) = \gamma\sigma\mathbf{G} + \mathbf{E}$. $\qquad\qquad\qquad \triangleright$ Where $\mathbf{G} = \mathrm{diag}(g_1, \ldots, g_n), g_i \leftarrow \mathcal{N}(0, 1)$.

5: $\widetilde{\mathbf{X}} = \mathbf{A} + \widetilde{\mathbf{V}} + \mathbf{E}^{(+)}$.

Output: Hermitian perturbed matrix $\widetilde{\mathbf{X}}$.

Ensures: See Proposition D.2.

Algorithm 5: REGULARIZE.

We can now use the Wegner estimate to get a minimum singular value bound for diagonal shifts over a grid, defined as follows.

Definition D.2 (Grid). *A 1-d grid in the real line is a set of s points defined as*

$$\mathrm{grid}(l, r, h) = \{l + jh | j \in \mathbb{Z}_{\geq 0}, l + jh \leq r\}.$$

The cardinality of a grid $\mathsf{g} = \mathrm{grid}(l, r, h)$ is denoted as $|\mathsf{g}|$.

The following Proposition 3.2 summarizes the properties of Algorithm 5, REGULARIZE, that we will use to perform efficient eigenvalue counting queries to compute the spectral gap.

Proposition D.2. *Let $\mathbf{A}$ with $\|\mathbf{A}\| \leq 1$ be a Hermitian matrix, $\gamma, \delta \in (0, 1/4)$ two given parameters, and $\widetilde{\mathbf{X}} \leftarrow$ REGULARIZE$(\mathbf{A}, \gamma, \delta)$. Let g be an arbitrary (but fixed) grid of points in $[-2, 2]$ with size $|\mathsf{g}| = T$. For every element $h_i \in \mathsf{g}$ consider the matrices $\mathbf{M}_i = h_i - \widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{M}} = h_i - \widetilde{\mathbf{X}} + \mathbf{E}_i$, where $\mathbf{E}_i$ denote the diagonal floating point error matrices induced by the shift. If the machine precision $\mathbf{u}$ satisfies*

$$\mathbf{u} \leq \frac{\gamma\delta}{32(c+2)nT\sqrt{\pi\ln(4n/\delta)}},$$

which translates to $O(\log(\frac{Tn}{\gamma\delta}))$ bits, then all the following hold simultaneously with probability $1 - 2\delta$:

$$\|\widetilde{\mathbf{X}}\| \leq 4/3, \quad \left|\lambda_i(\widetilde{\mathbf{X}}) - \lambda_i(\mathbf{A})\right| \leq \tfrac{9}{16}\gamma, \quad \sigma_{\min}(\widetilde{\mathbf{M}}_i) \geq \frac{\gamma\delta}{4nT\sqrt{4\pi\ln(4n/\delta)}}.$$

Proof. Let $\mathbf{X} = \mathbf{A} + \gamma\sigma\mathbf{G}$, where $\sigma\mathbf{G}$ is a diagonal matrix with independent elements from $\mathcal{N}(0, \sigma^2)$ where $\sigma = \frac{1}{2\sqrt{2\ln(4n/\delta)}}$. Let $\mathbf{B} = \mathbf{X}/\gamma = \mathbf{A}/\gamma + \sigma\mathbf{G}$. Denote by $E_{\mathbf{G}}$ the event $\|\sigma\mathbf{G}\| \leq \frac{1}{2}$. From a tail bound as in Lemma D.1, $\Pr[E_{\mathbf{G}}] \geq 1 - \delta$. Conditioning on $E_{\mathbf{G}}$, we obtain $\|\mathbf{B}\| = \|\mathbf{A}/\gamma + \sigma\mathbf{G}\| \leq 1/\gamma + 1/2 := \beta$. This directly implies that $\Lambda(\mathbf{B}) \subseteq (-1/\gamma - 1/2, 1/\gamma + 1/2)$. Let $h_i' = h_i/\gamma$ for all $i = 1, \ldots, T$.

From the Wegner estimate (Proposition D.1), for every fixed interval I,

$$\Pr\left[|\Lambda(h' - \mathbf{B}) \cap I| \geq 1\right] \leq \sqrt{4\pi\ln(4n/\delta)}n|I|.$$

Setting I_i to be an interval centered at h_i' with size $|I_i| = \frac{\delta}{Tn\sqrt{4\pi\ln(4n/\delta)}}$, we have that the probability that I_i contains any eigenvalues is at most δ/T. Since there are no eigenvalues in I_i, this means that the smallest singular value of $h_i' - \mathbf{B}$ is at least half the size of I_i, i.e. $\sigma_{\min}(h_i' - \mathbf{B}) \geq |I_i|/2 = \frac{\delta}{2Tn\sqrt{4\pi\ln(4n/\delta)}}$. Hence, $\sigma_{\min}(h_i - \mathbf{X}) = \gamma\sigma_{\min}(h_i' - \mathbf{B}) \geq \frac{\gamma\delta}{2Tn\sqrt{4\pi\ln(4n/\delta)}}$ holds as well.

From Lemma D.1, if the diagonal Gaussian sampler is called with $p = \Theta(\log(\log(\frac{n}{\delta})))$ and $t = \log(1/\mathbf{u})$ then

$$\|\mathbf{E}\| \leq \gamma\eta_{c+2}\frac{1}{2},$$

$$\|\widetilde{\mathbf{V}}\| \leq \gamma(1 + \eta_{c+2})\frac{1}{2},$$

where $|\eta_{c+2}| \leq \frac{(c+2)\mathbf{u}}{1-(c+2)\mathbf{u}}$. The elements of the diagonal error matrix $\mathbf{E}^{(+)}$ in Algorithm 5 are bounded by

$$|\mathbf{E}_{i,i}^{(+)}| \leq \mathbf{u}|\mathbf{A}_{i,i} + \widetilde{\mathbf{V}}_{i,i}| \leq \mathbf{u}(\|\mathbf{A}\| + \|\widetilde{\mathbf{V}}\|) \leq \mathbf{u}\left(1 + \tfrac{1}{2}\gamma(1 + \eta_{c+2})\right).$$

Then

$$\begin{aligned}
\|\widetilde{\mathbf{X}} - \mathbf{X}\| &= \|\mathbf{E}^{(+)} + \mathbf{E}\| \\
&\leq \mathbf{u}\|\mathbf{A} + \widetilde{\mathbf{V}}\| + \tfrac{1}{2}\gamma\eta_{c+2} \\
&\leq \mathbf{u}\left(1 + \tfrac{1}{2}\gamma(1 + \eta_{c+2})\right) + \tfrac{1}{2}\gamma\eta_{c+2}.
\end{aligned}$$

If $\mathbf{u} \leq \frac{\epsilon}{2(c+2)}$ for some $\epsilon \in (0,1)$ then $|\eta_{c+2}| \leq \epsilon$ and since $\gamma < 1/2$ we also have that $\|\widetilde{\mathbf{X}} - \mathbf{X}\| \leq \epsilon$. Recall that $\|\mathbf{X}\| = \|\mathbf{A} + \gamma\sigma\mathbf{G}\| \leq 1 + \gamma/2 \leq 5/4$ and thus

$$\|\mathbf{E}_i\| \leq \mathbf{u}(|h_i| + \|\widetilde{\mathbf{X}}\|) \leq \mathbf{u}(|h_i| + \|\mathbf{X}\| + \epsilon) \leq \mathbf{u}(2 + (1 + \gamma/2) + \epsilon) \leq 5\mathbf{u} \leq 5\frac{\epsilon}{2(c+2)} \leq \epsilon,$$

assuming that $c \geq 1$. We now collect all the error bounds and apply Weyl's inequality to argue that

$$|\lambda_j(h_i - \mathbf{X}) - \lambda_j(h_i - \widetilde{\mathbf{X}} + \mathbf{E}_i)| \leq \|\mathbf{X} - \widetilde{\mathbf{X}}\| + \|\mathbf{E}_i\| \leq 2\epsilon.$$

The same holds for the singular values σ_j since the matrices are Hermitian. Thus if

$$\epsilon \leq \frac{\gamma\delta}{8Tn\sqrt{4\pi\ln(4n/\delta)}} \leq \sigma_{\min}(h_i - \mathbf{X})/4$$

then

$$\left|\sigma_{\min}(h_i - \widetilde{\mathbf{X}} + \mathbf{E}_i) - \sigma_{\min}(h_i - \mathbf{X})\right| \leq \sigma_{\min}(h_i - \mathbf{X})/2,$$

which implies

$$\sigma_{\min}(h_i - \widetilde{\mathbf{X}} + \mathbf{E}_i) \geq \sigma_{\min}(h_i - \mathbf{X})/2 = \frac{\gamma\delta}{4Tn\sqrt{4\pi\ln(4n/\delta)}}.$$

The aforementioned bound on ϵ also gives two (loose) bounds that are useful for the analysis of algorithms later:

$$\|\widetilde{\mathbf{X}}\| \leq \|\mathbf{X}\| + \epsilon \leq 5/4 + 1/32 \leq 4/3,$$

and also from Weyl's inequality

$$\left|\lambda_i(\mathbf{A}) - \lambda_i(\widetilde{\mathbf{X}})\right| \leq \|\mathbf{A} - \mathbf{X}\| + \|\mathbf{X} - \widetilde{\mathbf{X}}\| \leq \tfrac{\gamma}{2} + \tfrac{\gamma}{16} = \tfrac{9}{16}\gamma.$$

The requirement for the machine precision is

$$\mathbf{u} \leq \frac{\epsilon}{2(c+2)} \leq \frac{\gamma\delta}{32(c+2)Tn\sqrt{\pi\ln(4n/\delta)}},$$

which translates to

$$O\left(\log(\tfrac{Tn}{\gamma\delta})\right)$$

bits of precision (recall that we also required $p = \Theta(\log(\log(n/\delta)))$ for the exponent of the Gaussian sampler, but this does not affect the total number of bits).

For the complexity, we have n calls to $\mathsf{N}(p,t)$ and $O(n)$ scalar additions and multiplications thereafter, which accumulate to $O(n)$ total floating point operations using $O\left(\log(\tfrac{Tn}{\gamma\delta})\right)$ bits.

For the success probability, there are two types random events that we took into account: $E_{\mathbf{G}}$, which fails with probability at most δ, and the Wegner estimate for T points h_i in the grid g, where each one fails with probability at most δ/T. A union bound over all random events gives a success probability of at least $1 - \delta - T\delta/T = 1 - 2\delta$. $\qquad\square$

E Fast spectral gaps and eigenvalues with counting queries

The core of our methods is an algorithm to efficiently approximate spectral gaps using only "eigenvalue counting queries," avoiding an explicit (and expensive) diagonalization.

E.1 Counting eigenvalues

The main subroutine of our algorithm is used to count the eigenvalues that are smaller than a given threshold efficiently.

Lemma E.1. *Let $\widetilde{\mathbf{X}}$ be Hermitian, $\|\widetilde{\mathbf{X}}\| \leq 2$, and $h \in [-2,2]$ is a fixed point that can be exactly represented in floating point, i.e. $\mathbf{fl}(h) = h$. There exists an algorithm $\mathsf{COUNT}(\widetilde{\mathbf{X}}, h, \varepsilon)$ which takes as input $h, \widetilde{\mathbf{X}}$, and $\varepsilon \in (0, 4/199)$ with the guarantee that $\sigma_{\min}(h - \widetilde{\mathbf{X}}) \geq \varepsilon$, and it returns an integer $n_{(-)}$, which is the precise number of eigenvalues of $\widetilde{\mathbf{X}}$ that are smaller than h. The algorithm requires*

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{1}{\varepsilon}) + \log(\log(\tfrac{n}{\varepsilon}))\right)\right)$$

arithmetic operations using

$$O\left(\log(n)\log^3(\tfrac{1}{\varepsilon})\log(\tfrac{n}{\varepsilon})\right)$$

bits of precision.

Proof. Let $\mathbf{M} = h - \widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{M}} = h - \widetilde{\mathbf{X}} + \mathbf{E}$ be its floating point counterpart. As the shift only distorts the diagonal elements, $\|\mathbf{E}\| = \max_i |\mathbf{E}_{i,i}| \leq \mathbf{u}\max_i |h - \widetilde{\mathbf{X}}_{i,i}| \leq 4\mathbf{u}$. By assumption $\sigma_{\min}(\mathbf{M}) \geq \varepsilon$, hence if $\mathbf{u} \leq \varepsilon/8$ then Weyl's inequality implies $\sigma_{\min}(\mathbf{M} + \mathbf{E}) \geq \sigma_{\min}(\mathbf{M}) - \varepsilon/2 \geq \varepsilon/2$.

Assume that we want to approximate $n_{(-)}$ by the expression

$$\widetilde{n}_{(-)} = \mathsf{COUNT}(\widetilde{\mathbf{X}}, h, \varepsilon) = \mathrm{round}\left(\mathbf{fl}(\mathrm{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}}, \alpha_{\mathsf{SGN}}, \eta_{\mathsf{SGN}}, \epsilon_{\mathsf{SGN}})))\right),$$

for some $\alpha_{\mathsf{SGN}}, \eta_{\mathsf{SGN}}, \epsilon_{\mathsf{SGN}}$ (to be determined). For simplicity we will use $\mathsf{SGN}(\widetilde{\mathbf{M}})$ to denote $\mathsf{SGN}(\widetilde{\mathbf{M}}, \alpha_{\mathsf{SGN}}, \eta_{\mathsf{SGN}}, \epsilon_{\mathsf{SGN}})$. Assuming for now that SGN converged to the requested error ϵ_{SGN}, we know that $\| \operatorname{sgn}(\widetilde{\mathbf{M}}) - \mathsf{SGN}(\widetilde{\mathbf{M}}) \| \le \epsilon_{\mathsf{SGN}}$. To proceed recall that for any matrix $\mathbf{A}$

$$|\mathsf{fl}(\operatorname{tr}(\mathbf{A})) - \operatorname{tr}(\mathbf{A})| \le \frac{\log(n)\mathbf{u}}{1 - \log(n)\mathbf{u}} \|\mathbf{A}\|,$$

which can be achieved with a binary tree-type addition.

Now consider the error

$$
\begin{aligned}
|n_{(-)} - \widetilde{n}_{(-)}| &= \left| \operatorname{tr}(\operatorname{sgn}(\mathbf{M})) - \operatorname{tr}(\operatorname{sgn}(\widetilde{\mathbf{M}})) + \operatorname{tr}(\operatorname{sgn}(\widetilde{\mathbf{M}})) - \operatorname{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}})) + \operatorname{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}})) - \mathsf{fl}(\operatorname{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}}))) \right| \\
&\le \left| \operatorname{tr}(\operatorname{sgn}(\mathbf{M})) - \operatorname{tr}(\operatorname{sgn}(\widetilde{\mathbf{M}})) \right| + \left| \operatorname{tr}(\operatorname{sgn}(\widetilde{\mathbf{M}})) - \operatorname{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}})) \right| + \left| \operatorname{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}})) - \mathsf{fl}(\operatorname{tr}(\mathsf{SGN}(\widetilde{\mathbf{M}}))) \right| \\
&\le \left| \operatorname{tr}(\operatorname{sgn}(\mathbf{M})) - \operatorname{tr}(\operatorname{sgn}(\widetilde{\mathbf{M}})) \right| + n\epsilon_{\mathsf{SGN}} + \frac{\log(n)\mathbf{u}}{1 - \log(n)\mathbf{u}}(1 + \epsilon_{\mathsf{SGN}}) \\
&\le n \left\| \operatorname{sgn}(\mathbf{M}) - \operatorname{sgn}(\widetilde{\mathbf{M}}) \right\| + n\epsilon_{\mathsf{SGN}} + \frac{\log(n)\mathbf{u}}{1 - \log(n)\mathbf{u}}(1 + \epsilon_{\mathsf{SGN}}). \quad (19)
\end{aligned}
$$

It remains to bound $\| \operatorname{sgn}(\mathbf{M}) - \operatorname{sgn}(\widetilde{\mathbf{M}}) \|$. For this we can directly use Lemma B.1 where we set $\mathbf{S} = \mathbf{I}$, $\mathbf{H} = \widetilde{\mathbf{X}}$, and $\mu = h$. The lemma states that if $\|\mathbf{E}\| \le \epsilon_{\mathsf{shift}} \frac{|\lambda_{\min}(h - \widetilde{\mathbf{X}})|^2 \pi}{128}$ then $\| \operatorname{sgn}(h - \widetilde{\mathbf{X}}) - \operatorname{sgn}(h - \widetilde{\mathbf{X}} + \mathbf{E}) \| \le \epsilon_{\mathsf{shift}}$ for some $\epsilon_{\mathsf{shift}} \in (0, 1)$ that we can choose. By assumption $|\lambda_{\min}(h - \widetilde{\mathbf{X}})| \ge \varepsilon$, which means that

$$4\mathbf{u} \le \epsilon_{\mathsf{shift}} \frac{\varepsilon^2 \pi}{128n}$$

is sufficient to guarantee that

$$n \left\| \operatorname{sgn}(\mathbf{M}) - \operatorname{sgn}(\widetilde{\mathbf{M}}) \right\| \le \epsilon_{\mathsf{shift}}.$$

For SGN, we set $\eta_{\mathsf{SGN}} = \frac{\varepsilon}{4}$, $\alpha_{\mathsf{SGN}} = \frac{4-\varepsilon}{4+\varepsilon}$ and $\epsilon_{\mathsf{SGN}} = \frac{1}{8n}$. Note that α_{SGN} satisfies the requirement $99/100 < \alpha_{\mathsf{SGN}} < 1$ as long as $\varepsilon < 4/199$). Then based on Theorem A.4 SGN requires

$$N = O\left(\log(\tfrac{1}{1 - \alpha_{\mathsf{SGN}}}) + \log(\log(\tfrac{1}{\epsilon_{\mathsf{SGN}} \eta_{\mathsf{SGN}}})) \right) = O\left(\log(\tfrac{1}{\varepsilon}) + \log(\log(\tfrac{n}{\varepsilon})) \right)$$

iterations and

$$O\left(\log(n) \log^3(\tfrac{1}{1 - \alpha_{\mathsf{SGN}}}) \left(\log(\tfrac{1}{\epsilon_{\mathsf{SGN}}}) + \log(\tfrac{1}{\eta_{\mathsf{SGN}}}) \right) \right) = O\left(\log(n) \log^3(\tfrac{1}{\varepsilon}) \log(\tfrac{n}{\varepsilon}) \right)$$

bits of precision.

If we set $\epsilon_{\mathsf{shift}} = 1/8$, then the bound for $\mathbf{u}$ becomes

$$\mathbf{u} \le \frac{\epsilon^2 \pi}{32 \cdot 128n},$$

in which case

$$\frac{\log(n)\mathbf{u}}{1 - \log(n)\mathbf{u}}(1 + \epsilon_{\mathsf{SGN}}) \ll \frac{1}{8}.$$

Then (19) becomes

$$|n_{(-)} - \widetilde{n}_{(-)}| < \tfrac{1}{8} + \tfrac{1}{8} + \tfrac{1}{8} = 3/8 < 1/2,$$

which means that if we round to the closest integer then $\operatorname{round}(\widetilde{n}_{(-)}) = n_{(-)}$. Note that $\log(1/\mathbf{u}) = O\left(\log(\tfrac{n}{\epsilon}) \right)$ which is dominated by the bit requirements of SGN.

$\square$

E.2 Approximate midpoint and gap

Having an efficient way to count eigenvalues smaller than a threshold, we show how to approximate spectral gaps. We can now prove Proposition 3.1, which we restate below for readability.

Proposition E.1. *Problem 3.1 can be solved in time* $O\left(\log(\frac{1}{\epsilon \, \mathrm{gap}_k})q(\frac{1}{\epsilon \, \mathrm{gap}_k})\right).$

Proof. We start with $\gamma_0 = 1/4$, and make a grid $\mathbf{g} = \mathrm{grid}(-1, 1+\gamma, \gamma)$ which consists of the points $\{-1, -3/4, -2/4, -1/4, 0, 1/4, 2/4, 3/4, 1, 5/4\}$ (a useful note for later is that all the numbers in the grid can be exactly represented in floating point). Let $\mathrm{count}(h_j)$ denote the number of γ_0-distorted values λ'_i that are smaller than h_j, for some $h_j \in \mathbf{g}, j = 1, \ldots, 10$. Query all $h_j \in \mathbf{g}$ for $\mathrm{count}(h_j)$. Set $j_k = \arg\min_j\{h_j \,|\, \mathrm{count}(h_j) \geq k\}$ and $j_{k+1} = \arg\min_j\{h_j \,|\, \mathrm{count}(h_j) \geq k+1\}$. Now $\lambda'_k \in [h_{j_k-1}, h_{j_k}]$ and $\lambda'_{k+1} \in [h_{j_{k+1}-1}, h_{j_{k+1}}]$. The points in the grid are equispaced so each interval has length γ. Since by definition $\lambda'_k \in [\lambda_k - \gamma_0, \lambda_k + \gamma_0]$, then $\lambda_k \in [h_{j_k-1} - \gamma_0, h_{j_k} + \gamma_0] = [h_{j_k-2}, h_{j_k+1}]$. Similarly for λ_{k+1}. We have now restricted both λ_k and λ_{k+1} inside some intervals of length $3\gamma_0$. Let us denote those intervals as $I_k = [h_{j_k-2}, h_{j_k+1}]$ and $I_{k+1} = [h_{j_{k+1}-2}, h_{j_{k+1}+1}]$. In the next step we halve γ_0 to $\gamma_1 = 1/8$. We now create two grids, one inside I_k and one inside I_{k+1}: $\mathbf{g}_k = \mathrm{grid}(h_{j_k-2}, h_{j_k+1} + \gamma_1, \gamma_1)$ and similar for $\mathbf{g}_{k+1}$. Each of the new grids has exactly 8 points and each point can be exactly represented in floating point. As in the previous iteration, we query all the 8 points h_j of $\mathbf{g}_k$ for $\mathrm{count}(h_j)$, and we pick a new $j_k = \arg\min\{h_j \,|\, \mathrm{count}(h_j) \geq k\}$, and we do the same for j_{k+1}. There are a total of 16 queries. As before, we have now restricted λ_k and λ_{k+1} inside two new intervals I_k and I_{k+1} where $|I_k| = |I_{k+1}| = 3\gamma_1$, which is half the size of the corresponding intervals of the previous iterations. If we set $\widetilde{\lambda}_k$ equal to the midpoint of I_k, then $|\widetilde{\lambda}_k - \lambda_k| \leq 3\gamma/2$. The same for λ_{k+1}. We keep repeating the same procedure by halving γ_i in each step i.

It remains to find a proper termination condition. Let us denote $\widetilde{\mu}_k = \frac{\widetilde{\lambda}_k + \widetilde{\lambda}_k}{2}$ and $\widetilde{\mathrm{gap}}_k = \widetilde{\lambda}_k - \widetilde{\lambda}_{k+1}$, where $\widetilde{\lambda}_k$ and $\widetilde{\lambda}_{k+1}$ are as above. Then, after m steps,

$$\widetilde{\mu}_k = \mu_k \pm 3\gamma_m/2,$$
$$\widetilde{\mathrm{gap}}_k = \mathrm{gap}_k \pm 3\gamma_m.$$

Thus, $\gamma_m \approx \epsilon \, \mathrm{gap}_k /3$ is a sufficient terminating criterion. We don't know gap_k, but we can use $\widetilde{\mathrm{gap}}_k$ instead, i.e., $\gamma \leq \epsilon \frac{\widetilde{\mathrm{gap}}_k - 3\gamma_m}{3}$ is also a sufficient, which gives a quantifiable termination criterion $\gamma_m \leq \epsilon \widetilde{\mathrm{gap}}_k/6$. But since $\widetilde{\mathrm{gap}}_k \geq \mathrm{gap}_k - 3\gamma_m$, then the terminating criterion will be reached in the worst case when

$$\gamma_m \leq \epsilon \frac{\mathrm{gap}_k - 3\gamma_m}{6} \Rightarrow \gamma_m \leq \frac{\epsilon \, \mathrm{gap}_k}{9},$$

meaning that the algorithm will halt and return an ϵ-accurate $\widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k$ in at most $O\left(\log(\frac{9}{\gamma_m})\right) = O\left(\log(\frac{1}{\epsilon \, \mathrm{gap}_k})\right)$ iterations.

Each h_j-query at iteration i costs $O(\mathrm{polylog}(1/\gamma_i))$ by assumption, and there are exactly 16 h_j-queries in each iteration, which gives a total query cost of $O(\mathrm{polylog}(1/\gamma_i))$ per iteration, which is maximized for the smallest $\gamma_m = \Theta(\epsilon \, \mathrm{gap}_k)$. $\square$

We can now describe the algorithm GAP that computes the k-th gap and the midpoint of a Hermitian definite pencil.

Theorem E.1 (GAP, Restatement of Theorem 3.1). *Let* $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}^n_{++}$ *and* $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\| \leq 1$, *which define a Hermitian definite pencil* $(\mathbf{H}, \mathbf{S})$. *Given* $k \in [n-1]$, *accuracy* $\epsilon \in (0,1)$, *and failure probability* $\delta \in (0, 1/2)$, *there exists an algorithm*

$$\widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k \leftarrow \mathsf{GAP}(\mathbf{H}, \mathbf{S}, k, \epsilon, \delta)$$

which returns $\widetilde{\mu}_k = \mu_k \pm \epsilon \, \mathrm{gap}_k$ *and* $\widetilde{\mathrm{gap}}_k = (1 \pm \epsilon) \, \mathrm{gap}_k$, *where* $\mu_k = \frac{\lambda_k + \lambda_{k+1}}{2}$ *and* $\mathrm{gap}_k = \lambda_k - \lambda_{k+1}$. *The algorithm requires*

$$O\left(T_{\mathsf{MM}}(n) \log(\frac{1}{\delta \epsilon \, \mathrm{gap}_k}) \log(\frac{1}{\epsilon \, \mathrm{gap}_k})\right)$$

arithmetic operations using $O\left(\log(n)\left(\log^4(\frac{n}{\delta\epsilon\,\mathrm{gap}_k})+\log(\kappa(\mathbf{S}))\right)\right)$ *bits, where* λ_i *are the eigenvalues of* $(\mathbf{H},\mathbf{S})$. *If* $\kappa(\mathbf{S})$ *is unknown, additional* $O(T_{\mathsf{MM}}(n)\log(\frac{n\kappa(\mathbf{S})}{\delta})\log(\kappa(\mathbf{S})))$ *floating point operations and* $O(\log(n)\log^4(\frac{n\kappa(\mathbf{S})}{\delta}))$ *bits are suficient to compute it with Corollary* E.1.

Proof. We first set $\gamma_0 = 1/8$ and call $\widetilde{\mathbf{H}} = \mathsf{REDUCE}\left(\mathbf{H},\mathbf{S},\frac{\gamma_0}{4}\right)$. From Proposition C.3, REDUCE requires $O(T_{\mathsf{MM}}(n))$ floating point operations using $O(\log(n)\log(\kappa(\mathbf{S})) + \log(1/\gamma_0))$ bits, and it returns a matrix $\widetilde{\mathbf{H}}$ that satisfies $|\lambda_i(\widetilde{\mathbf{H}}) - \lambda_i(\mathbf{H},\mathbf{S})| \le \frac{\gamma_0}{4}$.

We then call $\widetilde{\mathbf{X}} \leftarrow \mathsf{REGULARIZE}(\widetilde{\mathbf{H}}, \frac{\gamma_0}{2}, \delta_0)$, where $\delta_0 = \delta/2$ is the initial failure probability. From Proposition 3.2, for all i it holds that $|\lambda_i(\widetilde{\mathbf{X}}) - \lambda_i(\widetilde{\mathbf{H}})| \le 9\gamma_0/16$. Therefore, all the eigenvalues, which initially lie in $[-1,1]$, are distorted by at most γ_0.

Next, recall the counting query model of Proposition 3.1: in the first step we construct a grid $\mathbf{g} = \mathrm{grid}(-1, 1+\gamma_0, \gamma_0)$, where $|\mathbf{g}| = 10$. From Proposition 3.2, the regularization ensures that for every $h_j \in \mathbf{g}$ it holds that $\sigma_{\min}(h_j - \widetilde{\mathbf{X}} + \mathbf{E}) \ge \varepsilon_0$ for $\varepsilon_0 = \frac{\gamma_0\delta_0}{8|\mathbf{g}|n\sqrt{\pi\ln(4n/\delta_0)}}$ (this is included in the success probability of $1-\delta_0$ of Proposition 3.2). Then we execute $\mathsf{COUNT}(\widetilde{\mathbf{X}}, h_j, \varepsilon_0)$ for every h_j. We continue by halving at each step both γ and δ, constructing the corresponding grids as per Proposition 3.1, and counting eigenvalues over the grid. In each iteration after the first one, we keep track of two grids $\mathbf{g}$ with size $|\mathbf{g}| = \Theta(1)$ each, and therefore we can ignore $|\mathbf{g}|$ in the complexity.

After a total of $m = O\left(\log(\frac{1}{\epsilon\,\mathrm{gap}_k})\right)$ iterations, we have that $\gamma_m = \Theta(\epsilon\,\mathrm{gap}_k)$ and $\delta_m = \Theta(\delta_0\epsilon\,\mathrm{gap}_k)$. Invoking Lemma E.1, and plugging in the value of ε_m, each call to COUNT during the entire algorithm costs at most

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\frac{1}{\varepsilon_m}) + \log(\log(\frac{n}{\varepsilon_m}))\right)\right) = O\left(T_{\mathsf{MM}}(n)\log(\frac{n}{\gamma_m\delta_m})\right)$$

arithmetic operations using $O\left(\log(n)\log^3(\frac{1}{\varepsilon_m})\log(\frac{n}{\varepsilon_m})\right) = O\left(\log(n)\log^4(\frac{n}{\gamma_m\delta_m})\right)$ bits. The last iteration where $\gamma_m = \Theta(\epsilon\,\mathrm{gap}_k)$ and $\delta_m = \Theta(\delta_0\epsilon\,\mathrm{gap}_k)$ is the most expensive, which gives a total of

$$O\left(T_{\mathsf{MM}}(n)\log(\frac{n}{\delta_0\epsilon\,\mathrm{gap}_k}) \cdot m\right) = O\left(T_{\mathsf{MM}}(n)\log(\frac{n}{\delta_0\epsilon\,\mathrm{gap}_k})\log(\frac{1}{\epsilon\,\mathrm{gap}_k})\right)$$

arithmetic operations and

$$O\left(\log(n)\log^4(\frac{n}{\delta_0\epsilon\,\mathrm{gap}_k})\right)$$

bits. The failure probability is δ_0 in the first iteration and halved at each subsequent iteration, in which case a union bound converges to $2\delta_0 = \delta$.

To be able to call REDUCE in every step, we need to approximate $\kappa(\mathbf{S})$ in order to set the machine precision appropriately. For this we can use Corollary E.1, which is detailed in the next section, and it returns $\widetilde{\kappa} \in \Theta(\kappa(\mathbf{S}))$ with probability $1-\delta$. It requires $O(T_{\mathsf{MM}}(n)\log(\frac{n\kappa(\mathbf{S})}{\delta})\log(\kappa(\mathbf{S})))$ floating point operations using $O(\log(n)\log^4(\frac{n\kappa(\mathbf{S})}{\delta}))$ bits. $\qquad\square$

E.3 Singular values, singular gaps, and condition number

The gap finder can be extended to compute eigenvalues and singular values and singular gaps or arbitrary matrices instead of eigenvalue gaps of Hermitian matrices. The following algorithm in Proposition E.2 computes the k-th singular value of a matrix, for arbitrary $k \in [n]$, using counting queries.

Proposition E.2 (SIGMAK). *Given a matrix* $\mathbf{A} \in \mathbb{C}^{m\times n}, n \ge m$ *with* $\|\mathbf{A}\| \le 1$, *an integer* $k \in [n]$ *such that* $\mathrm{rank}(\mathbf{A}) \ge k$, *an accuracy* $\epsilon \in (0,1)$, *and failure probability* $\delta \in (0,1)$, *there exists an algorithm*

$$\widetilde{\sigma}_k \leftarrow \mathsf{SIGMAK}(\mathbf{A}, k, \epsilon, \delta),$$

which returns a value $\widetilde{\sigma}_k \in (1 \pm \epsilon)\sigma_k(\mathbf{A})$ with probability at least $1 - \delta$. The algorithm executes

$$O\left(T_{\mathsf{MM}}(n)\left(\tfrac{m}{n} + \log(\tfrac{n}{\delta\epsilon\sigma_k})\right)\log(\tfrac{1}{\epsilon\sigma_k})\right)$$

floating point operations and requires $O\left(\log(n)\log^4(\tfrac{n}{\delta\epsilon\sigma_k})\right)$ bits.

Notes: *If $m > n$ then we can apply the algorithm on $\mathbf{A}^\top$. If $\|\mathbf{A}\| > 1$, we can use Theorem A.7 to approximate $\|\mathbf{A}\|$ and scale accordingly. The $\mathrm{rank}(\mathbf{A}) \geq k$ assumption can potentially be omitted by a more sophisticated counting query strategy.*

Proof. We will solve the problem with counting queries, similar to Theorem 3.1, halving γ and δ in every iteration. We start with $\gamma_0 = 1/8$ and $\delta_0 = \delta/2$. First, we divide $\mathbf{A}$ by two, to ensure that $\|\mathbf{A}\| \leq 1/2$. Then we construct $\widetilde{\mathbf{A}} = \mathsf{HERM}(\mathsf{MM}(\mathbf{A}, \mathbf{A}^*)) = \mathbf{A}\mathbf{A}^*/4 + \mathbf{E}^{\mathsf{MM}}$. To perform this rectangular multiplication in practice, we partition $\mathbf{A}$ in m/n blocks of size $n \times n$ each, perform the individual multiplications and sum the results. We can ensure that $\|\mathbf{E}^{\mathsf{MM}}\| \leq \gamma_0/4$ if we use $O(\log(\tfrac{n}{\gamma_0}))$ bits. Then from Weyl's inequality $|\lambda_k(\widetilde{\mathbf{A}}) - \lambda_k(\mathbf{A}\mathbf{A}^*/4)| \leq \gamma_0/4$. We then call $\widetilde{\mathbf{X}} \leftarrow \mathsf{REGULARIZE}(\widetilde{\mathbf{A}}, \gamma_0, \delta_0)$. Conditioning on success of Proposition 3.2, for all i it holds that $|\lambda_i(\widetilde{\mathbf{X}}) - \lambda_i(\widetilde{\mathbf{A}})| \leq 9\gamma_0/16$, and therefore $|\lambda_i(\widetilde{\mathbf{X}}) - \lambda_i(\mathbf{A}\mathbf{A}^*/4)| \leq \gamma_0$.

Let $\lambda'_i = \lambda_i(\widetilde{\mathbf{X}})$ and $\lambda_i = \lambda_i(\mathbf{A}\mathbf{A}^*/4)$. Since all the λ_i are non-negative, then all the $\lambda'_i \in [-\gamma_0, 1 + \gamma_0]$. We thus construct grid $\mathbf{g} = \mathrm{grid}(0, 1 + \gamma_0, \gamma_0) = \{0, 1/8, 2/8, 3/8, \ldots, 1, 9/8\}$. We now need to perform counting queries similar to Theorem 3.1. Since we conditioned on success of Proposition 3.2, $\mathsf{REGULARIZE}$ ensures that for every $h_j \in \mathbf{g}$ it holds that $\sigma_{\min}(h_j - \widetilde{\mathbf{X}} + \mathbf{E}_{h_j}) \geq \varepsilon_0$ for $\varepsilon_0 = \frac{\gamma_0\delta_0}{8|\mathbf{g}|n\sqrt{\pi\ln(4n/\delta_0)}}$. Then we execute $C_j \leftarrow \mathsf{COUNT}(\widetilde{\mathbf{X}}, h_j, \varepsilon_0)$ for every $h_j \in \mathbf{g}$ (in total $\Theta(1)$ calls to COUNT). We set j_k to be the smallest j such that $C_j \geq k$. Now $\lambda'_1 \in [h_{j_k}, h_{j_k+1}]$. Since $\lambda'_k \in [\lambda_k - \gamma_0, \lambda_k + \gamma_0]$, then $\lambda_k \in [h_{j_1-1} - \gamma_0, h_{j_k} + \gamma_0] = [h_{j_k-2}, h_{j_k+1}] := I_k$, where $|I_k| = 3\gamma_0$ (we can always ignore the negative portion of I_k, if there is any).

In the next step we halve γ_0 to $\gamma_1 = 1/8$, and we create another grid inside I_k, $\mathbf{g}_1 = \mathrm{grid}(h_{j_k-2}, h_{j_k+1} + \gamma_1, \gamma_1)$. The new grid has at most 8 points. We query all the points h_j of $\mathbf{g}_1$ for $C_j \leftarrow \mathsf{COUNT}(\widetilde{\mathbf{X}}, h_j, \epsilon_1)$, and we pick a new $j_1 = \arg\min\{h_j | C_j \geq k\}$. Now λ_k lies inside a new interval I_k with $|I_k| = 3\gamma_1$, which is half the size of the interval of the previous iteration. If we set $\widetilde{\lambda}_k$ equal to the midpoint of I_k (which is always positive), then $|\widetilde{\lambda}_k - \lambda_k| \leq 3\gamma/2$. We keep repeating the same procedure by halving γ_i and δ_i in each step i.

After m steps we have that

$$\widetilde{\lambda}_k = \lambda_k \pm 3\gamma_m/2.$$

A sufficient termination criterion is $\gamma_m \approx \epsilon\lambda_k$. Instead of λ_k (which is unknown) we use $\widetilde{\lambda}_k$ instead. Then $\gamma \leq \epsilon\frac{\widetilde{\lambda}_k - 3\gamma_m}{3}$ is also a sufficient, which gives a termination criterion $\gamma_m \leq \epsilon\widetilde{\lambda}_k/6$ that we can actually calculate. In turn, this termination criterion will be reached in the worst case when

$$\gamma_m \leq \epsilon\frac{\lambda_k - 3\gamma_m/2}{6} \Rightarrow \gamma_m \leq \frac{\epsilon\lambda_k}{9},$$

meaning that the algorithm will halt and return an ϵ-accurate λ_k in at most $m = O\left(\log(\tfrac{9}{\gamma_m})\right) = O\left(\log(\tfrac{1}{\epsilon\lambda_k})\right)$ iterations.

Note that in each iteration we make a total of $\Theta(1)$ calls to COUNT. From Lemma E.1, if we plug in the corresponding ε_i, each call to COUNT during the entire algorithm costs at most

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n}{\gamma_i\delta_i})\right)$$

arithmetic operations using $O\left(\log(n)\log^4(\tfrac{n}{\gamma_i\delta_i})\right)$ bits. The cost is maximized in the last iteration where $\gamma_m = \Theta(\epsilon\lambda_k)$ and $\delta_m = \Theta(\delta_0\epsilon\lambda_k)$, which gives

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n}{\delta_0\epsilon\lambda_k}) \cdot m\right) = O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n}{\delta_0\epsilon\lambda_k})\log(\tfrac{1}{\epsilon\lambda_k})\right)$$

arithmetic operations and

$$O\left(\log(n)\log^4(\tfrac{n}{\delta_0\epsilon\lambda_k})\right)$$

bits. As in Theorem 3.1, since the failure probability is halved in each iteration, a union bound converges to $2\delta_0 = \delta$.

Now we have that $|\lambda_k - \widetilde{\lambda}_k| \le \epsilon\lambda_k$. Recall that $\sigma_k(\mathbf{A}) = \sqrt{\lambda_k(\mathbf{A}\mathbf{A}^*)} = 2\sqrt{\lambda_k}$, since we computed $\mathbf{A}\mathbf{A}^*$ and divided by two. If we set $\widetilde{\sigma}_k = 2\sqrt{\widetilde{\lambda}_k}$ then

$$(1-\epsilon)\frac{\sigma_k^2}{4} \le \frac{\widetilde{\sigma}_k^2}{4} \le (1+\epsilon)\frac{\sigma_k^2}{4} \Rightarrow (1-\epsilon)\sigma_k \le \widetilde{\sigma}_k \le (1+\epsilon)\sigma_k,$$

where the last holds since $\sqrt{1+\epsilon} \le 1+\epsilon$ and $1-\epsilon \le \sqrt{1-\epsilon}$ for $\epsilon \in (0,1)$. Note that we lazily assumed that we computed $\widetilde{\sigma}_k = 2\sqrt{\widetilde{\lambda}_k}$ exactly, which does not hold, the square root introduces a machine precision error. The calibration is left as an exercise. If we replace λ_k with $\sigma_k^2/4$ and $\delta_0 = \delta/2$ in the complexity bounds we get the desired

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n}{\delta\epsilon\sigma_k})\log(\tfrac{1}{\epsilon\sigma_k})\right)$$

arithmetic operations and $O\left(\log(n)\log^4(\tfrac{n}{\delta\epsilon\sigma_k})\right)$ bits. $\qquad\square$

We now have a tool for the condition number.

Corollary E.1 (COND). *Let $\mathbf{A} \in \mathbb{H}_{++}^n$. Given $\delta \in (0,1/2)$, we can compute $\widetilde{\kappa}$ such that $\kappa(\mathbf{A}) \le \widetilde{\kappa} \le 32\kappa(\mathbf{A})$, with an algorithm $\widetilde{\kappa} \leftarrow \mathsf{COND}(\mathbf{A},\delta)$ in*

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n\kappa(\mathbf{A})}{\delta})\log(\kappa(\mathbf{A})))\right)$$

using $O\left(\log(n)\log^4(\tfrac{n\kappa(\mathbf{A})}{\delta})\right)$ bits of precision with probability at least $1-\delta$.

Proof. We first compute $\widetilde{\Sigma} \in [0.9\|\mathbf{A}\|, 2\|\mathbf{A}\|]$ using Corollary A.1 with parameter $\delta/3$ in $O(n^2\log(n)\log(1/\delta))$ floating point operations using $O(\log(n))$ bits of precision. The algorithm succeeds with probability at least $1 - \frac{2\delta}{3}$. We then scale $\mathbf{A}' \leftarrow \mathbf{A}/M$ where M is the smallest power of two that is larger than $4\widetilde{\Sigma}$. This implies $\frac{1}{16} \le \|\mathbf{A}'\| \le \frac{1}{3.6}$, and also $\sigma_{\min}(\mathbf{A}') \in \left[\frac{\sigma_{\min}(\mathbf{A})}{16\|\mathbf{A}\|}, \frac{\sigma_{\min}(\mathbf{A})}{3.6\|\mathbf{A}\|}\right] = \left[\frac{1}{16\kappa(\mathbf{A})}, \frac{1}{3.6\kappa(\mathbf{A})}\right]$. Thus, it suffices to approximate $\sigma_{\min}(\mathbf{A}')$ and scale it to obtain the desired approximation for $\kappa(\mathbf{A})$.

We now call $\widetilde{\sigma}'_{\min} \leftarrow \mathsf{SIGMAK}(\mathbf{A}', 1, 1/2, \delta/3)$, which succeeds probability $1 - \delta/3$ and returns $\widetilde{\sigma}'_{\min} \in (1 \pm \frac{1}{2})\sigma_{\min}(\mathbf{A}')$. It requires

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\delta\sigma_{\min}(\mathbf{A}')})\log(\tfrac{1}{\sigma_{\min}(\mathbf{A}')})\right)\right) = O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n\kappa(\mathbf{A})}{\delta})\log(\kappa(\mathbf{A}))\right)$$

arithmetic operations and

$$O\left(\log(n)\log^4(\tfrac{n}{\delta\sigma_{\min}(\mathbf{A}')})\right) = O\left(\log(n)\log^4(\tfrac{n\kappa(\mathbf{A})}{\delta})\right)$$

bits. Then we can set $\widetilde{\kappa} = \frac{1}{\widetilde{\sigma}'_{\min}} \in [\kappa(\mathbf{A}), 32\kappa(\mathbf{A})]$. The success probability is $1 - \frac{2\delta}{3} - \frac{\delta}{3} = 1 - \delta$. $\quad\square$

E.4 Proof of Theorem 1.1

We now have all the prerequisites to prove that Algorithm 1 provides the guarantees of our main Theorem 1.1, which we restate for readability.

Theorem E.2 (Restatement of Theorem 1.1). *Let $(\mathbf{H},\mathbf{S})$ be a Hermitian definite pencil of size n, with $\|\mathbf{H}\|, \|\mathbf{S}^{-1}\| \le 1$, and $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$ its eigenvalues. There exists an algorithm*

$$\widetilde{\Pi}_k \leftarrow \mathsf{PROJECTOR}(\mathbf{H}, \mathbf{S}, k, \epsilon),$$

which takes as input **H**, **S**, *an integer* $1 \leq k \leq n-1$, *an error parameter* $\epsilon \in (0, 1)$ *and returns a matrix* $\widetilde{\mathbf{\Pi}}_k$ *such that*

$$\Pr\left[\left\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\right\| \leq \epsilon\right] \geq 1 - 1/n,$$

where $\mathbf{\Pi}_k$ *is the true spectral projector on the invariant subspace that is associated with the* k *smallest eigenvalues. The algorithm executes*

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}) + \log(n\kappa(\mathbf{S}))\log(\kappa(\mathbf{S})) + \log\left(\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)\right)\right)$$

floating point operations, using

$$O\left(\log(n)\left(\log^4(\tfrac{n}{\mathrm{gap}_k}) + \log^4(n\kappa(\mathbf{S})) + \log^3(\tfrac{1}{\epsilon\,\mathrm{gap}_k})\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)\right)$$

bits of precision, where $\kappa(\mathbf{S}) = \|\mathbf{S}\|\|\mathbf{S}^{-1}\|$ *and* $\mathrm{gap}_k = \lambda_{k+1} - \lambda_k$. *Internally, the algorithm needs to generate a total of at most* $\widetilde{O}(n)$ *standard normal floating point numbers using additional* $O(\log(\log(n)))$ *bits. In the regular case, when* $\mathbf{S} = \mathbf{I}$, *the* $O(\log(n\kappa(\mathbf{S}))\log(\kappa(\mathbf{S})))$ *term in the arithmetic complexity and the* $O(\log^4(n\kappa(\mathbf{S})))$ *term in the number of bits are removed.*

Proof. We first compute $\widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k, \widetilde{\kappa}$ which satisfy the requirements of Proposition 2.1 using COND and GAP. Afterwards we can use them to call PURIFY to obtain the spectral projector $\widetilde{\mathbf{\Pi}}_k$ such that $\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\| \leq \epsilon$. We first call COND$(\mathbf{S}, \frac{1}{4}, \frac{1}{2n})$, which, from Corollary E.1, requires

$$O\left(T_{\mathsf{MM}}(n)\log(n\kappa(\mathbf{S}))\log(\kappa(\mathbf{S})))\right)$$

arithmetic operations and

$$O\left(\log(n)\log^4(n\kappa(\mathbf{S}))\right)$$

bits. Then, given the result of COND, we call GAP$(\mathbf{H}, \mathbf{S}, k, \frac{1}{8}, \frac{1}{2n})$ which requires

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}))\right)$$

arithmetic operations and

$$O\left(\log(n)\log^4(\tfrac{n}{\mathrm{gap}_k}) + \log(\kappa(\mathbf{S}))\right)$$

bits. They both succeed at the same time with $1 - 1/n$ probability. PURIFY$(\mathbf{H}, \mathbf{S}, \widetilde{\mu}_k, \widetilde{\mathrm{gap}}_k, \widetilde{\kappa}, \epsilon)$ requires

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{1}{\mathrm{gap}_k}) + \log(\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k}))\right)\right)$$

floating point operations and

$$O\left(\log(n)\log^3(\tfrac{1}{\mathrm{gap}_k})\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)$$

bits. The total arithmetic complexity is therefore

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}) + \log(n\kappa(\mathbf{S}))\log(\kappa(\mathbf{S})) + \log\left(\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)\right)\right)$$

and the bit requirement is

$$O\left(\log(n)\left(\log^4(\tfrac{n}{\mathrm{gap}_k}) + \log^4(n\kappa(\mathbf{S})) + \log^3(\tfrac{1}{\mathrm{gap}_k})\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\mathrm{gap}_k})\right)\right).$$

Notice that the ϵ term appears only inside $\log\log$ in the arithmetic complexity and in only one of the $\log$ factors of the number of bits. $\qquad\square$

F DFT background

Density Functional Theory (DFT) [88], which was awarded the Nobel prize in Chemistry in 1998, is considered as one of the most common methods to perform electronic structure calculations thanks to its *ab initio* character. That means that no input besides the initial atomic structure is required to determine the relaxed geometry, energy levels, and the electron structure of a given material, in their bulk or nanostructured form. The main idea behind DFT is to describe the system and its properties by the electron density only. DFT calculations are widely used in industry and in academia to predict, for example, the properties of novel materials [102] or to optimize the performance of batteries [69], solar cells [142], or nanoelectronic devices [85]. Many scientific libraries implementing DFT algorithms have been developed, which persistently occupy supercomputing clusters and receive up to tens of thousands of citations annually at the time of this writing [89, 59, 127, 74]. Despite the success of DFT calculations, important theoretical aspects of their underlying algorithms remain unclear.

Assume a system that is composed of a set of $n_a \in \mathbb{N}$ atoms positioned inside a fixed three-dimensional domain. The Kohn-Sham equation in real-space describes the electronic wave function $\psi(\mathbf{r})$ of the system

$$\mathcal{H}(\mathbf{r})\psi(\mathbf{r}) = E\psi(\mathbf{r}), \tag{20}$$

where $\mathbf{r} = (x, y, z)$ is any position in the domain. Here $\mathcal{H}(r)$ is the so-called single-particle Hamiltonian operator of the system, which can be written in the following form:

$$\mathcal{H}(\mathbf{r}) = -\frac{1}{2}\nabla^2 + E_{ion}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|}d\mathbf{r}' + E_{xc}(\mathbf{r}),$$

where the first term is the kinetic term, the second term accounts for electron-ion interactions, the third term represents the electron-electron interactions, which are solved at the Hartree level through Poisson's equation, and the fourth term is the so-called exchange-correlation term. The electron density $n(\mathbf{r})$ is the main observable of interest in DFT. It is related to the solutions E_i and $\psi_i(\mathbf{r})$ of Eq. (20) through

$$n(\mathbf{r}) = \sum_i f(E_i; E_F)|\psi_i(\mathbf{r})|^2, \tag{21}$$

where E_F is the Fermi level of the system, E_i is the energy level (eigenvalue) corresponding to the wave function $\psi_i(\mathbf{r})$, and $f(E_i; E_F)$ is the occupation term. At zero temperature $(T = 0K)$, $f(E_i; E_F)$ is equal to one for occupied states $(E_i < E_F)$ and zero otherwise.[3] The E_F, E_i, and ψ_i are the unknown quantities for which there exists no analytical solution, except for some special cases. Approximate numerical solutions can be sought after expanding the wave functions into a suitable basis. In this case, the ψ_i's are approximated by a set of $n = \Theta(n_a)$ basis functions $\chi_j(\mathbf{r}), j \in [n]$, typically localized around atomic positions.[4] Common basis sets that lead to efficient algorithms to solve the eigenvalue problem of Eq. (20) include Linear Combination of Atomic Orbitals (LCAO) [32], Gaussian-Type Orbitals (GTO) [22], or Maximally Localized Wannier Functions (MLWF) [101]. In this context, the wave functions take the form of linear combinations of the n localized orbitals. In particular, for all $i \in [n]$

$$\psi_i(\mathbf{r}) = \sum_{i=1}^{n} c_{ij}\chi_j(\mathbf{r}), \tag{22}$$

where c_{ij} are the (unknown) complex coefficients. This expansion can also be written in matrix notation as

$$\mathbf{y}(\mathbf{r}) = \begin{pmatrix} \psi_1(\mathbf{r}) \\ \psi_2(\mathbf{r}) \\ \vdots \\ \psi_n(\mathbf{r}) \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{pmatrix} \begin{pmatrix} \chi_1(\mathbf{r}) \\ \chi_2(\mathbf{r}) \\ \vdots \\ \chi_n(\mathbf{r}) \end{pmatrix} = \mathbf{C}\mathbf{x}(\mathbf{r}), \tag{23}$$

---

[3]The definition of $f(E; E_F)$ can vary to allow for partial occupations, i.e. at nonzero temperature $T > 0$ it can be replaced by the Fermi-Dirac distribution $1 - \frac{1}{1+\exp((E_i - E_F)/k_B T)}$, where k_B is Boltzmann's constant. This case is not considered in this work.

[4]A common alternative is to approximate the electronic wave function using Plane Waves. In this work we focus on localized basis sets.

where $\mathbf{y}(\mathbf{r}) \in \mathbb{C}^n$ and $\mathbf{x}(\mathbf{r}) \in \mathbb{C}^n$ are (complex) vectors and $\mathbf{C}$ is a $n \times n$ matrix. By expanding the wave functions into LCAO, GTO, or MLWF basis, the Hamiltonian operator becomes a matrix

$$\mathbf{H}_{i,j} = \langle \chi_i | \mathcal{H} | \chi_j \rangle = \int \chi_i^*(\mathbf{r}) \mathcal{H}(\mathbf{r}) \chi_j(\mathbf{r}) dr.$$

Similarly, an overlap matrix $\mathbf{S}$ must be introduced

$$\mathbf{S}_{i,j} = \langle \chi_i | \chi_j \rangle = \int \chi_i^*(\mathbf{r}) \chi_j(\mathbf{r}) dr,$$

which is equal to the identity matrix $\mathbf{I}$ if the basis elements are orthogonal (e.g. $\mathbf{S} = \mathbf{I}$ for MLWF). The entries of $\mathbf{H}$ and $\mathbf{S}$ represent the interaction between two basis elements that can be located on the same or on different atoms. Furthermore, due to the complex conjugate property of the inner product, both $\mathbf{H}$ and $\mathbf{S}$ are Hermitian matrices. Moreover, $\mathbf{S}$ is positive-definite ([103, Chapter 3]). Putting everything together, the expansion coefficients c_{ij} are obtained by solving the following generalized eigenvalue problem as already formulated in Eq. (1):

$$\mathbf{HC} = \mathbf{SC\Lambda}.$$

The so-called density matrix $\mathbf{P}$ can be derived from the $\mathbf{C}$ and $\mathbf{\Lambda}$ solutions

$$\mathbf{P} = \mathbf{C} f(\mathbf{\Lambda}; E_F) \mathbf{C}^*. \tag{24}$$

The density matrix possesses interesting properties. The matrix $\mathbf{PS}$ is a projector matrix and therefore it is idempotent, i.e. $\mathbf{PS} = (\mathbf{PS})^2$. Moreover, it can be used to derive the electron density at any position r

$$n(\mathbf{r}) = \sum_{i,j \in n} \mathbf{P}_{ij} \chi_i(\mathbf{r}) \chi_j^*(\mathbf{r}) = \mathbf{x}^*(\mathbf{r}) \mathbf{P} \mathbf{x}(\mathbf{r}), \tag{25}$$

where $\mathbf{x}(\mathbf{r}) \in \mathbb{C}^n$ is a vector such that $\mathbf{x}_i(\mathbf{r}) = \chi_i(\mathbf{r})$, i.e. $\mathbf{x}(\mathbf{r})$ contains the values of all the basis functions at position r. Given these basic definitions, we next give an overview of existing algorithms.

F.1 Linear scaling methods

Because of the strong localization of the $\chi_i(\mathbf{r})$'s, the (i, j) interactions rapidly decay with the distance between the atoms and can be discarded beyond a pre-defined cut-off radius. As a consequence, $\mathbf{H}$ and $\mathbf{S}$ are often banded matrices with sometimes a sparse band. It should also be noted that at equilibrium, the number of negatively charged electrons present in the system (which corresponds to the integral of $n(\mathbf{r})$ over space) must exactly compensate the number of positively charged protons Z_p. This allows for the computation of the Fermi level: from Eq. (25) and the definition of $\mathbf{S}$, it follows that $\text{trace}(\mathbf{PS}) = Z_p$, which means that one can perform a binary search on the Fermi level E_F until $\text{trace}(\mathbf{PS}) = Z_p$ is satisfied. These observation has led to the development of so-called linear scaling methods, which aim to exploit the sparsity of the matrices to obtain faster solutions [143, 57, 86, 87, 56, 62, 63]. Typically, the density matrix and the Fermi level are iteratively computed, until all required constraints are satisfied, i.e., the density matrix should be idempotent and $\text{trace}(\mathbf{PS}) = Z_p$. This can be done, for example, by approximating the matrix sign function using a Newton iteration [136]. To optimize performance, heuristics such as truncating matrix elements with negligible magnitude are applied, at the cost of decreasing the solution accuracy. Empirical evidence shows that such methods tend to scale nearly linearly to the system size, however, they do not exhibit provable theoretical guarantees. Similar methods have been studied based on finite-differences, which aim to approximate the Fermi distribution with polynomial expansions [145, 144]. Finally, the work of [65, 137] offer the possibility to achieve lower complexities than $O(n^3)$ for both the eigenvalues and eigenvectors of structured matrices, by leveraging the fast multipole method. However, an end-to-end stability analysis of those algorithms remains open. Indeed, we are not aware of any end-to-end analysis with provable approximation guarantees that has lower than $O(n^3)$ worst-case complexity for any of the aforementioned algorithms.

F.2 Density matrix

We can directly apply Theorem 1.1 to compute density matrices in DFT.

The following Theorem summarizes the analysis of Algorithm 6 for the density matrix.

<div style="border:1px solid black; padding:10px;">

<div align="center">DENSITY</div>

Input: Hermitian definite pencil $\mathbf{H} \in \mathbb{H}^n$, $\mathbf{S} \in \mathbb{H}_{++}^n$, number of occupied orbitals k.

Requires: $\|\mathbf{H}\| \leq 1$, $\|\mathbf{S}^{-1}\| \leq 1$, $k \in [n-1]$.

Algorithm: $\widetilde{\mathbf{P}} \leftarrow \text{DENSITY}(\mathbf{H}, \mathbf{S}, k, \epsilon)$.

1: $\widetilde{\mathbf{\Pi}} \leftarrow \text{PROJECTOR}(\mathbf{H}, \mathbf{S}, k, \frac{\epsilon}{32})$.

2: $\mathbf{S}_{\text{INV}} \leftarrow \text{INV}(\mathbf{S})$. $\qquad\qquad\qquad\qquad\qquad\qquad \triangleright \mathbf{S}_{\text{INV}} = \mathbf{S}^{-1} + \mathbf{E}_1^{\text{INV}}$.

3: $\widetilde{\mathbf{P}} \leftarrow \text{HERM}\left(\text{MM}(\widetilde{\mathbf{\Pi}}, \text{MM}(\mathbf{S}_{\text{INV}}, \widetilde{\mathbf{\Pi}}^*))\right)$. $\qquad \triangleright \widetilde{\mathbf{P}} = \widetilde{\mathbf{\Pi}}\left(\mathbf{S}_{\text{INV}}\widetilde{\mathbf{\Pi}}^* + \mathbf{E}_2^{\text{MM}}\right) + \mathbf{E}_3^{\text{MM}}$.

4: **return** $\widetilde{\mathbf{P}}$.

Output: Approximate density matrix $\widetilde{\mathbf{P}}$.

Ensures: $\|\widetilde{\mathbf{P}} - \mathbf{P}\| \leq \epsilon$ with probability at least $1 - O(1/n)$.

</div>

<div align="center">**Algorithm 6:** DENSITY.</div>

Theorem F.1 (DENSITY). *Given a Hermitian definite pencil* $(\mathbf{H}, \mathbf{S})$ *with* $\|\mathbf{H}\|, \|\mathbf{S}\| \leq 1$, *an integer* $k \in [n-1]$ *denoting the number of occupied states in the system, and* $\epsilon \in (0, 1)$, *Algorithm 6 returns a matrix* $\widetilde{\mathbf{P}} \leftarrow \text{DENSITY}(\mathbf{H}, \mathbf{S}, k, \epsilon)$ *such that*

$$\|\widetilde{\mathbf{P}} - \mathbf{P}\| \leq \epsilon,$$

where $\mathbf{P}$ *is the true density matrix of the system and succeeds with probability at least* $1 - O(1/n)$. *The floating point arithmetic complexity is*

$$O\left(T_{\text{MM}}(n)\left(\log(\tfrac{n}{\text{gap}_k})\log(\tfrac{1}{\text{gap}_k}) + \log(n\kappa(\mathbf{S}))\log(\kappa(\mathbf{S}))\right)\right),$$

and it requires

$$O\left(\log(n)\left(\log^4(\tfrac{n}{\text{gap}_k}) + \log^4(n\kappa(\mathbf{S})) + \log^3(\tfrac{1}{\epsilon\,\text{gap}_k})\log(\tfrac{\kappa(\mathbf{S})}{\epsilon\,\text{gap}_k})\right)\right)$$

bits of precision.

Proof. In the first step of the algorithm we compute $\widetilde{\mathbf{\Pi}}$ such that

$$\|\mathbf{\Pi} - \widetilde{\mathbf{\Pi}}\| \leq \tfrac{\epsilon}{32} := \epsilon_{\mathbf{\Pi}},$$

where $\mathbf{\Pi}$ is the spectral projector on the invariant subspace of the definite pencil associated with the k smallest eigenvalues.

Then we invert $\mathbf{S}$, and the final step of the algorithm computes $\widetilde{\mathbf{P}}$. Unrolling all the computations, $\widetilde{\mathbf{P}}$ can be written as $\mathbf{\Pi}\mathbf{S}^{-1}\mathbf{\Pi}^* + \mathbf{E}_{\mathbf{P}}$ where $\mathbf{E}_{\mathbf{P}}$ is a Hermitian error matrix (yet to be detailed). It is not hard to verify that $\mathbf{\Pi}\mathbf{S}^{-1}\mathbf{\Pi}^*$ is equal to $\mathbf{P}$, i.e. the true density matrix.

It thus remains to bound $\|\mathbf{E}_{\mathbf{P}}\|$. We can directly force $\|\mathbf{E}_1^{\text{INV}}\| \leq \epsilon_{\mathbf{\Pi}}\|\mathbf{S}^{-1}\| \leq \epsilon_{\mathbf{\Pi}}$ by setting $\mathbf{u} \leq \epsilon_{\mathbf{\Pi}}\frac{1}{\mu_{\text{INV}}(n)\kappa(\mathbf{S})^{c_{\text{INV}}\log(n)}}$. Then from line 5 of Algorithm 6

$$\begin{aligned}
\widetilde{\mathbf{P}} &= \widetilde{\mathbf{\Pi}}\mathbf{S}_{\text{INV}}\widetilde{\mathbf{\Pi}}^* + \widetilde{\mathbf{\Pi}}\mathbf{E}_2^{\text{MM}} + \mathbf{E}_3^{\text{MM}} \\
&= (\mathbf{\Pi} + \mathbf{E}_{\mathbf{\Pi}})\left(\mathbf{S}^{-1} + \mathbf{E}_1^{\text{INV}}\right)(\mathbf{\Pi} + \mathbf{E}_{\mathbf{\Pi}})^* + (\mathbf{\Pi} + \mathbf{E}_{\mathbf{\Pi}})\mathbf{E}_2^{\text{MM}} + \mathbf{E}_3^{\text{MM}} \\
&= \mathbf{\Pi}\mathbf{S}^{-1}\mathbf{\Pi}^* + \mathbf{\Pi}\mathbf{E}_1^{\text{INV}}\mathbf{\Pi}^* + \mathbf{\Pi}\left(\mathbf{S}^{-1} + \mathbf{E}_1^{\text{INV}}\right)\mathbf{E}_{\mathbf{\Pi}}^* + \mathbf{E}_{\mathbf{\Pi}}\left(\mathbf{S}^{-1} + \mathbf{E}_1^{\text{INV}}\right)\mathbf{\Pi}^* + \ldots \\
&\qquad \ldots + \mathbf{E}_{\mathbf{\Pi}}\left(\mathbf{S}^{-1} + \mathbf{E}_1^{\text{INV}}\right)\mathbf{E}_{\mathbf{\Pi}}^* + (\mathbf{\Pi} + \mathbf{E}_{\mathbf{\Pi}})\mathbf{E}_2^{\text{MM}} + \mathbf{E}_3^{\text{MM}}.
\end{aligned}$$

We bound each term as follows.

(i) $\|\mathbf{\Pi}\mathbf{E}_1^{\text{INV}}\mathbf{\Pi}^*\| \leq \|\mathbf{E}_1^{\text{INV}}\| \leq \epsilon_{\mathbf{\Pi}}$,

<div align="center">60</div>

(ii) $\left\|\mathbf{\Pi}\left(\mathbf{S}^{-1} + \mathbf{E}_1^{\mathsf{INV}}\right)\mathbf{E}_{\mathbf{\Pi}}^*\right\| \leq (1 + \epsilon_{\mathbf{\Pi}}) \cdot \epsilon_{\mathbf{\Pi}} \leq 2\epsilon_{\mathbf{\Pi}},$

(iii) $\left\|\mathbf{E}_{\mathbf{\Pi}}\left(\mathbf{S}^{-1} + \mathbf{E}_1^{\mathsf{INV}}\right)\mathbf{E}_{\mathbf{\Pi}}^*\right\| \leq (1 + \epsilon_{\mathbf{\Pi}}) \cdot \epsilon_{\mathbf{\Pi}}^2 \leq 2\epsilon_{\mathbf{\Pi}}^2,$

(iv) $\|\mathbf{E}_2^{\mathsf{MM}}\| \leq \mathbf{u}\mu(n)\|\mathbf{S}_{\mathsf{INV}}\|\|\widetilde{\mathbf{\Pi}}^*\| \leq \mathbf{u}\mu(n)\left(\|\mathbf{S}^{-1}\| + \|\mathbf{E}_1^{\mathsf{INV}}\|\right)(1 + \|\mathbf{E}_{\mathbf{\Pi}}\|) \ll \epsilon_{\mathbf{\Pi}}(1 + \epsilon_{\mathbf{\Pi}})(1 + \epsilon_{\mathbf{\Pi}}) \leq 4\epsilon_{\mathbf{\Pi}},$

(v) $\|\mathbf{E}_3^{\mathsf{MM}}\| \leq \mathbf{u}\mu(n)\|\widetilde{\mathbf{\Pi}}\|\|\mathbf{S}_{\mathsf{INV}}\widetilde{\mathbf{\Pi}}^* + \mathbf{E}_2^{\mathsf{MM}}\| \ll \epsilon_{\mathbf{\Pi}}(1 + \epsilon_{\mathbf{\Pi}})\left[(1 + \epsilon_{\mathbf{\Pi}})(1 + \epsilon_{\mathbf{\Pi}}) + 4\epsilon_{\mathbf{\Pi}}\right] \leq \epsilon_{\mathbf{\Pi}} \cdot 2 \cdot (4 + 4\epsilon_{\mathbf{\Pi}}) \leq 16\epsilon_{\mathbf{\Pi}}.$

Putting everything together we have that

$$\left\|\mathbf{P} - \widetilde{\mathbf{P}}\right\| \leq \epsilon_{\mathbf{\Pi}} + 2\epsilon_{\mathbf{\Pi}} + 2\epsilon_{\mathbf{\Pi}}^2 + 4\epsilon_{\mathbf{\Pi}} + 16\epsilon_{\mathbf{\Pi}} \leq 25\epsilon_{\mathbf{\Pi}} = \epsilon\frac{25}{32} \leq \epsilon.$$

Note that the aforementioned value of $\mathbf{u}$ is already dominated by the requirement of PROJECTOR. The same holds for the arithmetic complexity. Therefore, the arithmetic complexity, the success probability, and the number of required bits, similar to those of Theorem 1.1 up to a constant factor. □

F.3 Electron density queries

Having a good approximation of the density matrix, we can use it to query for the electron density at any point of interest. A direct application of Theorem F.1 gives the following.

Corollary F.1. *Given a position* $\mathbf{r}$ *in the atomic domain, let* $\mathbf{x}$ *be a vector* $\mathbf{x} = (\chi_1(\mathbf{r}) \quad \chi_2(\mathbf{r}) \quad \ldots \quad \chi_n(\mathbf{r}))$, *where* χ_i *are the basis functions that were used to construct the Hamiltonian and overlap matrices* $\mathbf{H}, \mathbf{S}$. *Let* $\widetilde{\mathbf{P}} \leftarrow \mathsf{DENSITY}(\mathbf{H}, \mathbf{S}, \epsilon, k)$ *where* k *is the number of occupied states in the atomic system, and let* $\mathbf{P}$ *be the true density matrix. Let* $n(\mathbf{r}) = \mathbf{x}^*\mathbf{P}\mathbf{x}$ *be the true electron density at* $\mathbf{r}$ *and* $\widetilde{n}(\mathbf{r}) = \mathbf{x}^*\widetilde{\mathbf{P}}\mathbf{x}$. *Then as long as* DENSITY *succeeds,*

$$|n(\mathbf{r}) - \widetilde{n}(\mathbf{r})| \leq 8\epsilon\|\mathbf{x}\|^2.$$

Proof. First we recall that $\epsilon \in (0, 1/12)$ and note that $\|\widetilde{\mathbf{P}}\| \leq \|\mathbf{P}\| + \epsilon \leq 1 + \epsilon$. The quadratic form $\mathbf{x}^*\widetilde{\mathbf{P}}\mathbf{x}$ is computed in two steps: $\mathbf{y} \leftarrow \mathbf{fl}(\widetilde{\mathbf{P}}\mathbf{x})$ and $\widetilde{n}(\mathbf{r}) = \mathbf{fl}(\mathbf{x}^*\mathbf{y})$. From [70], Eq. (3.12) and Lemma 6.6, we know that the matrix-vector product satisfies $\mathbf{y} = \widetilde{\mathbf{P}}\mathbf{x} + \mathbf{e}$ where $\|\mathbf{e}\| = \|\widetilde{\mathbf{P}}\mathbf{x} - \mathbf{y}\| \leq \sqrt{n}\eta_n\|\widetilde{\mathbf{P}}\|\|\mathbf{x}\|$, and $\eta_n := \frac{n\mathbf{u}}{1-n\mathbf{u}}$. For the chosen $\mathbf{u}$ we have $\sqrt{n}\eta_n \leq 2\epsilon$. From the same equations, the subsequent dot product satisfies $\widetilde{n}(\mathbf{r}) = \mathbf{fl}(\mathbf{x}^*\mathbf{y}) = \mathbf{x}^*\mathbf{y} + \epsilon$ where $|\epsilon| \leq \eta_n\|\mathbf{x}\|\|\mathbf{y}\| \leq \eta_n\|\mathbf{x}\|\left(\|\widetilde{\mathbf{P}}\mathbf{x}\| + \|\mathbf{e}\|\right) \leq \eta_n\|\widetilde{\mathbf{P}}\|\|\mathbf{x}\|^2(1 + \sqrt{n}\eta_n) \leq 4\epsilon\|\mathbf{x}\|^2$. Then $|n(\mathbf{r}) - \widetilde{n}(\mathbf{r})| = |\mathbf{x}^*\mathbf{P}\mathbf{x} - (\mathbf{x}^*\mathbf{y} + \epsilon)| = |\mathbf{x}^*\mathbf{P}\mathbf{x} - \mathbf{x}^*\widetilde{\mathbf{P}}\mathbf{x} - \mathbf{x}^*\mathbf{e} - \epsilon| \leq |\mathbf{x}^*\mathbf{P}\mathbf{x} - \mathbf{x}^*\widetilde{\mathbf{P}}\mathbf{x}| + |\mathbf{x}^*\mathbf{e}| + |\epsilon| \leq 2\epsilon\|\mathbf{x}\|^2 + 2\epsilon\|\mathbf{x}\|^2 + 4\epsilon\|\mathbf{x}\|^2 = 8\epsilon\|\mathbf{x}\|^2$. □

This can be generalized for many different points $\mathbf{r}$. If N is the number of those points and $N = \Theta(n)$ (which is typically the case in applications), then one can stack the vectors $\mathbf{x}(\mathbf{r}_1), \mathbf{x}(\mathbf{r}_2), \ldots, \mathbf{x}(\mathbf{r}_N)$ as the rows of a matrix $\mathbf{X}$ and use fast matrix multiplication to compute all $n(\mathbf{r}_i)$ simultaneously, instead of querying each $n(\mathbf{r}_i)$ one-by-one.

If N is asymptotically larger than the system size, i.e., $N = O(\text{poly}(n))$, the query complexity can be reduced by applying the Johnson-Lindenstrauss (JL) lemma [78], at the cost of a polynomial dependence on the accuracy. In brief, since the matrix $\mathbf{PS}$ is an orthogonal projector, and therefore idempotent, it can be deduced that

$$\mathbf{x}^*\mathbf{P}\mathbf{x} = \mathbf{x}^*\mathbf{PSP}\mathbf{x} = \mathbf{x}^*\mathbf{PL}^{-*}\mathbf{L}^{-1}\mathbf{P}\mathbf{x} = \|\mathbf{x}^*\mathbf{PL}^{-*}\|^2,$$

where $\mathbf{L}$ is the lower triangular Cholesky factor of $\mathbf{S}^{-1}$. Then $\|\mathbf{x}^*\mathbf{PL}^{-*}\|^2$ can be approximated by $\|\mathbf{x}^*\mathbf{PL}^{-*}\mathbf{T}\|^2$, where $\mathbf{T}$ is a $n \times r$ sparse random sign matrix scaled by $1/\sqrt{r}$ [1], with $r = O(\log(N)/\epsilon^2)$ columns. The representation of $\mathbf{T}$ requires $O(n \times r) = O(n\log(N)/\epsilon^2)$ random bits. Alternative random matrix distributions satisfying the JL property, with varying sparsity, runtime,

and dimension requirements exist in the literature [3, 82, 110]. The value of r is worst-case optimal [93], however, tighter bounds have been studied in the literature [21, 126, 23, 49], thus potentially leading to more practical algorithms. The full analysis is omitted.

G Deflation and bit complexity of PCA

Since its introduction in the early twentieth century [116, 72], Principal Component Analysis is one of the most important tools in statistics, data science, and machine learning. It can be used, for example, to visualize data, to reduce dimensionality, or to remove noise in measurements; cf. [79, 45] for reviews on the vast bibliography. In its simplest formulation, given a (centered) data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the goal is to find a k-dimensional embedding $\mathbf{C}_k$, where $k < n$, that maximizes the sample variance, which can be written as an optimization problem

$$\mathbf{C}_k = \arg \max_{\mathbf{C}^\top \mathbf{C} = \mathbf{I}_{k \times k}} \text{tr}(\mathbf{C}^\top \mathbf{H} \mathbf{C}), \tag{26}$$

where $\mathbf{H} = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{n \times n}$ is the sample covariance. It can be shown that the solution $\mathbf{C}_k$ corresponds to the principal k singular vectors of $\mathbf{H}$, i.e. the ones that correspond to the largest k singular values. Evidently, since the sample covariance is always symmetric and positive semi-definite, this can be written as a Hermitian eigenvalue problem

$$\mathbf{H}\mathbf{C} = \mathbf{C}\mathbf{\Lambda},$$

(which is indeed a definite GEP as in Equation (1) with $\mathbf{S} = \mathbf{I}$). This way we can project the data in k dimensions by computing $\mathbf{X}\mathbf{C}_k$, preserving as much of the variance in k dimensions as possible. Classically, PCA can be solved by approximating the SVD of $\mathbf{H}$ via diagonalization.

G.1 Vanilla PCA

With this short introduction, we can now describe how to enhance the analysis of PCA algorithms by using our results for spectral projectors to obtain forward-error guarantees for all the SVD-related computations. Using our Theorem 1.1, the SIGMAK algorithm of Proposition E.2, and the DEFLATE algorithm of [19], as described in Algorithm 7, we can state the following result for the classic ("vanilla") PCA, which provides forward error guarantees in matrix multiplication time regardless how large k is. This is can serve as the backbone for more advanced algorithms that are analyzed in the next section, which typically first use random projections to reduce the matrix size to approximately $O(k \times k)$ and then perform the SVD computation on the smaller matrix.

Remark G.1. *For simplicity, we have assumed that the covariance matrix contains no errors. This might not be true if $\mathbf{H}$ is computed numerically from the data matrix $\mathbf{X}$. It is not hard to extend the algorithm to take this into account as well, and nothing changes in the analysis (except some negligible constant factors). To handle errors in the input matrix one can adapt the analysis from the next Section G.2.*

Theorem G.1 (PCA). *Let $\mathbf{H}$ be an $n \times n$ symmetric sample covariance matrix of a centered data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, i.e. $\mathbf{H} = \mathbf{X}^\top \mathbf{X}$, $\|\mathbf{H}\| \leq 1$, $k \in [n]$ is a target rank, and $\epsilon \in (0, 1)$ an accuracy parameter. Then we can compute a matrix $\widetilde{\mathbf{C}}_k$ with k columns such that $\|\mathbf{X} - \mathbf{X}\widetilde{\mathbf{C}}_k\widetilde{\mathbf{C}}_k^\top\| \leq (1 + \epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^\top\|$, where $\mathbf{C}_k \in \mathbb{R}^{n \times k}$ contains the top-k (right) singular vectors of $\mathbf{X}$ in*

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\sigma_{k+1}})\log(\tfrac{1}{\sigma_{k+1}}) + \log(\tfrac{n}{\text{gap}_k})\log(\tfrac{1}{\text{gap}_k}) + \log(\log(\tfrac{n}{\epsilon\sigma_{k+1}\,\text{gap}_k}))\right)\right)$$

arithmetic operations using $O\left(\log(n)\left(\log^4(\tfrac{n}{\epsilon\,\text{gap}_k}) + \log^4(\tfrac{n}{\sigma_{k+1}})\right) + \log(\tfrac{1}{\epsilon\sigma_{k+1}})\right)$ bits of precision, with probability at least $1 - O(1/n)$.

Proof. We first compute $\widetilde{\sigma}_{k+1} \leftarrow \mathsf{SIGMAK}(\mathbf{H}, k+1, \tfrac{1}{2}, \tfrac{1}{n})$ such that, from Proposition E.2, $\widetilde{\sigma}_{k+1} \in (1 \pm \tfrac{1}{2})\sigma_{k+1}(\mathbf{H})$ with probability $1 - 1/n$. It requires

$$O\left(T_{\mathsf{MM}}(n)\log(\tfrac{n}{\sigma_{k+1}})\log(\tfrac{1}{\sigma_{k+1}})\right)$$

62

PCA

Input: Sample Covariance Matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$ of a centered data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, target rank k, accuracy $\epsilon \in (0, 1)$.

Requires: $\|\mathbf{H}\| \leq 1$, $k \in [n-1]$.

Algorithm: $\widetilde{\mathbf{C}}_k \leftarrow \mathsf{PCA}(\mathbf{H}, k, \epsilon)$.

1: $\widetilde{\sigma}_k \leftarrow \mathsf{SIGMAK}(\mathbf{H}, k, \frac{1}{2}, \frac{1}{3n})$

2: $\epsilon' \leftarrow \epsilon \widetilde{\sigma}_k / 4$.

3: $\widetilde{\mathbf{\Pi}}_k \leftarrow \mathsf{PROJECTOR}(\mathbf{H}, \mathbf{I}, k, (\frac{\epsilon'^2}{20^3 n^4})^2)$.

4: $\widetilde{\mathbf{C}}_k \leftarrow \mathsf{DEFLATE}(\widetilde{\mathbf{\Pi}}_k, k, (\frac{\epsilon'^2}{20^3 n^4})^2, \epsilon')$.

5: **return** $\widetilde{\mathbf{C}}_k$.

Output: Approximate principal component matrix $\widetilde{\mathbf{C}}_k$.

Ensures: $\|\mathbf{X} - \mathbf{X}\widetilde{\mathbf{C}}_k \widetilde{\mathbf{C}}_k^\top\| \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^T\|$ with probability at least $1 - O(1/n)$.

Algorithm 7: PCA.

floating point operations and $O\left(\log(n)\log^4(\frac{n}{\sigma_{k+1}})\right)$ bits. Then we set $\epsilon' \leftarrow \epsilon\widetilde{\sigma}_{k+1}/4$. In the next step, we compute $\widetilde{\mathbf{\Pi}}_k \leftarrow \mathsf{PROJECTOR}(\mathbf{H}, \mathbf{I}, k, (\frac{\epsilon'^2}{20^3 n^4})^2)$, which, from Theorem 1.1, returns a spectral projector $\widetilde{\mathbf{\Pi}}_k$ that satisfies $\|\widetilde{\mathbf{\Pi}}_k - \mathbf{\Pi}_k\| \leq (\frac{\epsilon'^2}{20^3 n^4})^2$ with probability $1 - 1/n$. It requires

$$O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}) + \log(\log(\tfrac{n}{\epsilon'\,\mathrm{gap}_k}))\right)\right)$$
$$= O\left(T_{\mathsf{MM}}(n)\left(\log(\tfrac{n}{\mathrm{gap}_k})\log(\tfrac{1}{\mathrm{gap}_k}) + \log(\log(\tfrac{n}{\epsilon\sigma_{k+1}\,\mathrm{gap}_k}))\right)\right)$$

arithmetic operations and $O\left(\log(n)\log^4(\frac{n}{\epsilon'\,\mathrm{gap}_k})\right) = O\left(\log(n)\log^4(\frac{n}{\epsilon\sigma_{k+1}\,\mathrm{gap}_k})\right)$ bits. Then we use DEFLATE to compute the matrix $\mathbf{C}'_k \leftarrow \mathsf{DEFLATE}(\widetilde{\mathbf{\Pi}}_k, k, (\frac{\epsilon'^2}{20^3 n^4})^2, \epsilon')$. From Theorem A.5, DEFLATE succeeds with probability

$$1 - \frac{(20n)^3\sqrt{(\epsilon'^2/(20^3 n^4))^2}}{\epsilon'^2} = 1 - 1/n,$$

and it requires $O(T_{\mathsf{MM}}(n))$ floating point operations using $O(\log(n/\epsilon')) = O(\log(n/(\sigma_{k+1}\epsilon)))$ bits, and it internally generates $O(n^2)$ random complex normal variables. On success it returns $\mathbf{C}'_k \in \mathbb{C}^{n \times k}$ such that $\|\mathbf{C}_k - \mathbf{C}'_k\| \leq \epsilon'$, where $\mathbf{C}_k$ is a matrix whose columns form an orthonormal basis for $\mathbf{\Pi}_k$. We can then keep only the real part of $\mathbf{C}'_k$, and set $\widetilde{\mathbf{C}}_k \leftarrow Re(\mathbf{C}'_k)$. The spectral norm can only decrease by removing the imaginary part, therefore $\|\widetilde{\mathbf{C}}_k - \mathbf{C}_k\| \leq \epsilon'$ holds as well. Now we can write $\widetilde{\mathbf{C}}_k = \mathbf{C}_k + \mathbf{E}$ where $\|\mathbf{E}\| \leq \epsilon'$, which means that

$$\left\|\mathbf{X} - \mathbf{X}\widetilde{\mathbf{C}}_k\widetilde{\mathbf{C}}_k^\top\right\| = \left\|\mathbf{X} - \mathbf{X}(\mathbf{C}_k + \mathbf{E})(\mathbf{C}_k + \mathbf{E})^\top\right\|$$
$$\leq \left\|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^\top\right\| + \left\|\mathbf{X}\mathbf{E}\mathbf{C}_k^\top\right\| + \left\|\mathbf{X}\mathbf{C}_k\mathbf{E}^\top\right\| + \left\|\mathbf{X}\mathbf{E}\mathbf{E}^\top\right\|$$
$$\leq \left\|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^\top\right\| + 2\epsilon'\|\mathbf{X}\| + \epsilon'^2\|\mathbf{X}\|$$
$$\leq \left\|\mathbf{X} - \mathbf{C}_k\mathbf{C}_k^\top\mathbf{X}\right\| + 3\frac{\epsilon\widetilde{\sigma}_{k+1}}{4}$$
$$\leq (1 + \tfrac{9}{8}\epsilon)\left\|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^\top\right\|,$$

where in the last we used the fact that

$$\widetilde{\sigma}_{k+1}(\mathbf{H}) \leq 3\sigma_{k+1}(\mathbf{H})/2 = 3\sigma_{k+1}^2(\mathbf{X})/2 \leq 3\sigma_{k+1}(\mathbf{X})/2 = 3\|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^\top\|/2.$$

The result follows by rescaling ϵ by a constant and by summing together the individual algorithm complexities and bits. $\qquad\square$

63

G.2 Low-rank PCA

In many applications of PCA the parameter k is chosen to be small, i.e. $k \ll d$. This has led to an extensive research area on the so-called *low-rank approximation* algorithms for PCA. One of the earliest works which introduced this type of randomized low-rank approximations is [53, 54]. Some other landmark works in the field include the analysis of randomized PCA and low-rank approximation [100, 67, 120, 33, 34], and the pioneering Block-Krylov PCA of [106], which is essentially optimal in the matrix-vector query model [125]. The approximation accuracy of low-rank approximation-based PCA methods is often measured with respect to the spectral or the Frobenius norm error, i.e. the matrix $\widetilde{\mathbf{C}}_k$ that is returned should satisfy $\mathbf{C}_k^\top \mathbf{C}_k = \mathbf{I}_{k \times k}$ and:

$$\left\| \mathbf{X} - \mathbf{X}\widetilde{\mathbf{C}}_k\widetilde{\mathbf{C}}_k^\top \right\|_{\{2,F\}} \leq (1+\epsilon) \left\| \|\mathbf{X} - \mathbf{X}\mathbf{C}_k\mathbf{C}_k^\top\| \right\|_{\{2,F\}}. \tag{27}$$

Most of the aforementioned low-rank approximation type algorithms assume exact arithmetic. A subtle detail is that they often rely on the computation of an (exact) SVD of some smaller submatrix, which is not realistic due to the Abel-Ruffini theorem. This can be justified for practical reasons since SVD can be approximated in polynomial time to arbitrary accuracy using classical solvers [64]. Quoting [67]:

> *"Techniques for computing the SVD are iterative by necessity, but they converge so fast that we can treat them as finite for practical purposes."*

However, if one wants to rigorously prove forward error approximations and end-to-end complexity upper bounds, there is necessarily a dependence on the singular value gap that separates the principal invariant subspace from the rest of the spectrum (consider a small perturbation of the 2-by-2 identity matrix and $k = 1$ as a straightforward example).

Assuming an exact SVD algorithm, the seminal analysis of the Block-Krylov PCA of [106] can in fact provide per-singular vector guarantees, which are much stronger than the classical norm-wise bounds. Then arithmetic complexity of Block-Krylov PCA depends on $\text{poly}(\frac{k}{\sqrt{\epsilon}})$. It is favorable for coarse accuracy ϵ (i.e. when $\epsilon = \Theta(1)$) and small rank k. It is not suitable, however, for larger k. E.g., it can be the case in applications where we need to keep $k = n/20$ of the original dimensions. The same holds when higher accuracy is required, e.g. when $\epsilon = 1/\text{poly}(n)$. Then the complexity of these methods is already higher than standard eigensolvers.

In finite precision, the landscape is even less clear. At the time of this submission, the only work related to low-rank approximation PCA that we are aware of with end-to-end bit complexity bounds is [107] for the approximation of matrix functions applied on vectors. They prove that the Lanczos method can be stably applied to compute a vector $\mathbf{u}$ such that the quantity $\|\mathbf{A}\mathbf{u}\|$ approximates $\|\mathbf{A}\|$. This is a backward-approximate solution for the top-1 singular vector. Concurrently with this work, and independently, [80] analyzed the bit complexity of Block-Krylov PCA and achieved similar bounds as ours, albeit with different techniques. However, [80] did not describe how to compute the condition number which is required in order to adjust the machine precision. Our analysis covers the computation of all the involved parameters.

For general invariant subspaces ($k > 1$), consider one of the simplest randomized low-rank approximation algorithms, often referred to as "subspace iteration" [67] or "simultaneous iteration" [106]. The algorithm first samples a matrix $\mathbf{G} \in \mathbb{R}^{n \times l}$ with i.i.d standard normal elements, where $l = \Theta(k)$, and computes $\mathbf{Y} \leftarrow (\mathbf{A}\mathbf{A}^T)^q\mathbf{A}\mathbf{G}$, where typically $q \approx \log(\min\{m, n\})$. It then returns the $\mathbf{Q}$ factor from the economy QR factorization of $\mathbf{Y}$. It can be shown (see e.g. Corollary 10.10 of [67]) that

$$\|(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top)\mathbf{A}\| \lesssim \|\mathbf{A} - \mathbf{A}_k\|.$$

The returned matrix has slightly more than k columns, often referred to as "oversampling," but, importantly, there is no explicit SVD involved in the computation. Even in this case, floating point arithmetic already spoils the approximation guarantees: we cannot compute $\mathbf{Q}$ without rounding errors. Stability analysis of QR factorization is typically carried out in the backward-error sense [70, 39], which makes the analysis of low-rank approximation PCA algorithms even more complicated.

G.3 Bit complexity analysis of Block-Krylov PCA

We now analyze the seminal Block-Krylov iteration algorithm of [106], which is listed in Algorithm 8 for convenience (in exact arithmetic). For the floating point analysis the following methodology is used. The main result is stated in Theorem 4.2. We shall denote by $T_{\mathsf{MM}}(\mathbf{X}, q)$ the cost of multiplying $\mathbf{X}$ with a dense matrix with q columns with a numerically stable multiplication algorithm, like the one of Theorem A.2. Formally, for the rest of this section, we assume the following subroutine.

Definition G.1 (MMX). *Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be the input matrix of Algorithm 8. We assume a subroutine* $\mathsf{MMX}(\mathbf{B})$*, which takes as input a matrix $\mathbf{B}$ with k columns. It returns a matrix $\mathbf{C} \in \mathbb{R}^{m \times r}$ such that*

$$\|\mathbf{C} - \mathbf{X}\mathbf{B}\| \leq \mathbf{u}\operatorname{poly}(m, k)\|\mathbf{X}\|\|\mathbf{B}\|,$$

in a total of $T_{\mathsf{MMX}}(k)$ floating point operations, using at most $O(\log(1/\mathbf{u}))$ bits of precision. We also assume $\mathsf{MMX}^{\top}(\mathbf{B})$ which approximates the product $\mathbf{X}^{\top}\mathbf{B}$ with the same cost and approximation bounds.

The standard inner-product based algorithm as well as a block variant of the MM algorithm of Theorem A.2 satisfy this definition.

Block-Krylov Iteration (exact arithmetic)

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, target rank k, accuracy $\epsilon \in (0, 1)$.

Requires: Exact arithmetic.

Algorithm: $\mathbf{Z}_k \leftarrow$ Block-Krylov Iteration $(\mathbf{X}, k, \epsilon)$.

1: $q \leftarrow \Theta(\frac{\log(n)}{\sqrt{\epsilon}})$.
2: $\mathbf{G} \leftarrow \mathcal{N}(0, 1)^{n \times k}$.
3: $\mathbf{K} \leftarrow \left[\mathbf{X}\mathbf{G}, (\mathbf{X}\mathbf{X}^{\top})\mathbf{X}\mathbf{G}, \ldots, (\mathbf{X}\mathbf{X}^{\top})^q\mathbf{X}\mathbf{G}\right]$.
4: $\mathbf{Q}, \mathbf{R} \leftarrow$ Economy-QR$(\mathbf{K})$. $\triangleright \mathbf{Q} \in \mathbb{R}^{m \times qk}$.
5: $\mathbf{M} \leftarrow \mathbf{Q}^{\top}\mathbf{X}\mathbf{X}^{\top}\mathbf{Q} \in \mathbb{R}^{qk \times qk}$.
6: $\bar{\mathbf{U}}_k \leftarrow$ top-k singular vectors of $\mathbf{M}$.
7: **return** $\mathbf{Z}_k \leftarrow \mathbf{Q}\bar{\mathbf{U}}_k$.

Output: Approximate principal component matrix $\mathbf{Z}$.

Ensures: $\|\mathbf{X} - \mathbf{Z}_k\mathbf{Z}_k^{\top}\mathbf{X}\| \leq (1 + \epsilon)\|\mathbf{X} - \mathbf{X}_k\|$ with high probability (in exact arithmetic).

Algorithm 8: Block-Krylov Iteration (Alg. 2 of [106])

Step 1: Constructing the Block-Krylov matrix. We first observe that we can scale the matrix $\mathbf{G}$ in Algorithm 8 to have norm at most one, since this does not affect the Krylov basis. We assume that the (scaled) Gaussian matrix $\mathbf{G}$ in Algorithm 8 is given exactly. This is not realistic, but it greatly simplifies the analysis and it can be addressed by using the floating point Gaussian sampler of Definition D.1 to obtain a small error with high probability.

Given $\mathbf{G}$ and $\mathbf{X}$, we can construct the Krylov matrix $\mathbf{K}$ using $O(q)$ calls to $\mathsf{MMX}(\mathbf{G})$. The first call $\mathbf{X}_1 \leftarrow \mathsf{MMX}(\mathbf{G})$ returns a matrix $\mathbf{X}_1 = \mathbf{X}\mathbf{G} + \mathbf{E}_1$ where $\|\mathbf{E}_1\| \leq \mathbf{u} \cdot O(\operatorname{poly}(mk))\|\mathbf{X}\|\|\mathbf{G}\| \leq \mathbf{u} \cdot O(\operatorname{poly}(mk))$, since we assumed constant norms. If we keep performing multiplications recursively to build the Krylov matrix $\|\widetilde{\mathbf{K}}\|$, after $O(q)$ multiplications it holds that $\|\widetilde{\mathbf{K}} - \mathbf{K}\| \leq \mathbf{u} \cdot O(\operatorname{poly}(mkq))$. The total cost is $O(qT_{\mathsf{MMX}}(k))$ floating point operations. If $\mathbf{X}$ is sparse and we use the standard inner product-based algorithm, $T_{\mathsf{MMX}}(k) = O(k\operatorname{nnz}(\mathbf{X}))$. If $\mathbf{X}$ is dense and we use Theorem A.2 then $O(qmk^{\omega-2})$ floating point operations are sufficient.

Step 2: Condition number of the Block-Krylov matrix. So far we have approximated $\mathbf{K}$ by $\widetilde{\mathbf{K}} = \mathbf{K} + \mathbf{E}_{\mathbf{K}}$.

In the following steps we will need an approximation for the condition number of $\mathbf{K}$. We can get such an approximation using a variant of the COND algorithm (Corollary E.1), which internally uses

the SIGMAK algorithm of Proposition E.2 based on counting queries to approximate the condition number. In that case, we start with some arbitrary error $\|\mathbf{E_K}\|$, and we keep keep halving it until we have a sufficiently good approximation. At each iteration t, the algorithm executes

$$O\left(qT_{\mathsf{MMX}}(k)\right)$$

floating point operations to construct the Krylov matrix using $O(\mathrm{polylog}(m, q, \frac{1}{\epsilon_t})$ bits. At most $O(\log(\kappa(\mathbf{K})))$ iterations are required. The most expensive iteration is the last one, when $\epsilon_t \approx 1/\kappa(\mathbf{K})$, which gives a total cost of at most

$$O\left(q\log(\kappa(\mathbf{K}))T_{\mathsf{MMX}}(k)\right)$$

floating point operations using at most $O(\mathrm{polylog}(m, q, \kappa(\mathbf{K})))$ bits. It returns a value $\widetilde{\kappa} \in \Theta(\kappa(\mathbf{K}))$, and it succeeds with high probability.

Step 3: Computing a basis. The computation of the basis in Step 4 of Algorithm 8 is arguably the hardest part in the analysis of the Block-Krylov iteration. With similar arguments as in the proof of Theorem A.6, if we write the economy-QR factorizations $\widetilde{\mathbf{K}} = \mathbf{Q}_{\widetilde{\mathbf{K}}}\mathbf{R}_{\widetilde{\mathbf{K}}}$ and $\mathbf{K} = \mathbf{QR}$, there exists an orthogonal matrix $\boldsymbol{\Phi}_1$ such that $\mathbf{Q} - \mathbf{Q}_{\widetilde{\mathbf{K}}}\boldsymbol{\Phi}_1 = \mathbf{E}_1$ and $\|\mathbf{E}_1\| \leq \epsilon'$, for some ϵ'. This requires that $\|\widetilde{\mathbf{K}} - \mathbf{K}\| \leq \epsilon'\frac{1}{4}\frac{1}{\mathrm{poly}(mnqk)\kappa(\mathbf{K})}$, which implies a requirement

$$\mathbf{u} \leq \epsilon'\frac{c}{\mathrm{poly}(mq)\kappa(\mathbf{K})},$$

for some constant c. Instead of $\kappa(\mathbf{K})$ we can use $\widetilde{\kappa}$ from the previous Step 2. We can now use Theorem A.6 (which computes a basis via QR) on $\widetilde{\mathbf{K}}$ to approximate $\mathbf{Q}_{\widetilde{\mathbf{K}}}$ by the matrix

$$\widetilde{\mathbf{Q}} = \mathbf{Q}_{\widetilde{\mathbf{K}}}\boldsymbol{\Phi}_2 + \mathbf{E}_{\mathsf{QR}} = (\mathbf{Q}\boldsymbol{\Phi}_1 + \mathbf{E}_1)\boldsymbol{\Phi}_2 + \mathbf{E}_{\mathsf{QR}} = \mathbf{Q}\boldsymbol{\Phi}_1\boldsymbol{\Phi}_2 + \mathbf{E}_1\boldsymbol{\Phi}_2 + \mathbf{E}_{\mathsf{QR}}.$$

Let $\mathbf{E}_{\mathbf{Q}} = \mathbf{E}_1\boldsymbol{\Phi}_2 + \mathbf{E}_{\mathsf{QR}}$. We have that $\|\mathbf{E}_{\mathbf{Q}}\| = \|\mathbf{E}_1\boldsymbol{\Phi}_2 + \mathbf{E}_{\mathsf{QR}}\| \leq \epsilon' + \mathbf{E}_{\mathsf{QR}}$. From Theorem A.6 we can ensure that $\|\mathbf{E}_{\mathsf{QR}}\| \leq \epsilon'$ if we use $O(\log(\frac{mq\kappa(\widetilde{\mathbf{K}})}{\epsilon'}))$ bits. Thus, $\widetilde{\mathbf{Q}}$ is an approximate orthogonal basis for the range of the true Block-Krylov matrix $\mathbf{K}$.

To summarize, as long as $\mathbf{u} \leq \epsilon'\frac{c}{\mathrm{poly}(mq)\kappa(\mathbf{K})}$, which can be achieved by replacing $\kappa(\mathbf{K})$ with $\widetilde{\kappa}$ above, then we can compute $\widetilde{\mathbf{Q}} = \mathbf{Q}\boldsymbol{\Phi} + \mathbf{E}_{\mathbf{Q}}$, where $\mathbf{Q}$ is the true orthonormal basis from the QR factorization of $\mathbf{K}$, $\boldsymbol{\Phi}$ is a $qk \times qk$ orthogonal matrix, and $\|\mathbf{E}_{\mathbf{Q}}\| \leq \epsilon'$. The total cost is $O(m(qk)^{\omega-1})$ floating point operations for Theorem A.6, using $O\left(\log(\frac{mq\kappa(\mathbf{K})}{\epsilon'})\right)$ bits.

Step 4: Computing the reduced matrix. In Step 5, Algorithm 8 forms the matrix $\mathbf{M}$ to compute its top-k singular vectors. To analyze the computation of $\mathbf{M}$, the first observation is that $\mathbf{Q}$ in lines 4-7 can be replaced by any basis for the column space of $\mathbf{K}$. In particular, we replace it by $\widetilde{\mathbf{Q}}$. We then perform the multiplication in two steps: $\widetilde{\mathbf{M}}_1^\top = \mathsf{MMX}^\top(\widetilde{\mathbf{Q}})$, which returns $\widetilde{\mathbf{M}}_1 = \widetilde{\mathbf{Q}}^\top\mathbf{X} + \mathbf{E}_1^{\mathsf{MM}}$, where $\|\mathbf{E}_1^{\mathsf{MM}}\| \leq \mathbf{u} \cdot O(\mathrm{poly}(qknm))\|\widetilde{\mathbf{Q}}\|\|\mathbf{X}\| \in \mathbf{u} \cdot O(\mathrm{poly}(qm))$. Then we compute $\widetilde{\mathbf{M}} \leftarrow \mathsf{MM}(\widetilde{\mathbf{M}}_1, \widetilde{\mathbf{M}}_1^\top) = \widetilde{\mathbf{M}}_1\widetilde{\mathbf{M}}_1^\top + \mathbf{E}_2^{\mathsf{MM}}$ where $\|\mathbf{E}_2^{\mathsf{MM}}\| \leq \mathbf{u} \cdot O(\mathrm{poly}(qkn))\|\mathbf{M}_1\|^2 \in \mathbf{u} \cdot O(\mathrm{poly}(qkn))$, where the last is implied if $\mathbf{u}$ is sufficiently smaller than $1/\mathrm{poly}(qm)$. Putting everything together we can write

$$\widetilde{\mathbf{M}} = \boldsymbol{\Phi}_2^\top\boldsymbol{\Phi}_1^\top\mathbf{Q}^\top\mathbf{XX}^\top\mathbf{Q}\boldsymbol{\Phi}_1\boldsymbol{\Phi}_2 + \mathbf{E_M},$$

where $\mathbf{E_M}$ contains the errors in $\widetilde{\mathbf{Q}}$ and also the ones from the multiplication in floating point, and $\|\mathbf{E_M}\| \leq O(\|\mathbf{E}_{\mathbf{Q}}\|) \in O(\epsilon')$. ϵ' is as in Step 3. So far we have assumed $O(\log(\frac{mq\kappa(\widetilde{\mathbf{K}})}{\epsilon'}))$ bits.

Step 5: Spectral gap and midpoint. Now we have written $\widetilde{\mathbf{M}} = \boldsymbol{\Phi}^\top\mathbf{Q}^\top\mathbf{XX}^\top\mathbf{Q}\boldsymbol{\Phi} + \mathbf{E_M}$, where $\boldsymbol{\Phi} = \boldsymbol{\Phi}_2\boldsymbol{\Phi}_1$ is orthogonal and $\|\mathbf{E_M}\| \leq O(\epsilon')$ for some $\epsilon' \in (0, 1)$. We can use the counting queries in the spirit of Theorem 3.1 to compute the spectral gap and the midpoint of $\mathbf{M}$.

In particular, we start with some desired bound $\epsilon'_0 \in (0, 1)$ for $\|\mathbf{E_M}\|$ by setting the number of bits to $O(\log(\frac{qm\kappa(\widetilde{\mathbf{K}})}{\epsilon'_0}))$. At each iteration we repeat Steps 1, 3, and 4 (we do not need to compute

66

again the condition number of $\mathbf{K}$ in Step 2). After at most $m = O(\log(1/\operatorname{gap}_k(\mathbf{M})))$ iterations, the error ϵ'_m satisfies $\epsilon'_m = \Theta(\operatorname{gap}_k(\mathbf{M}))$, and we obtain two quantities $\widetilde{\operatorname{gap}}_k \in (1 \pm \frac{1}{8})\operatorname{gap}_k(\mathbf{M})$ and $\widetilde{\mu}_k \in \mu_k(\mathbf{M}) + \frac{1}{8}\operatorname{gap}_k(\mathbf{M})$ with high probability. In each iteration we need to construct $\widetilde{\mathbf{M}}$ using Steps 1, 3, and 4 with the specified number of bits.

Step 1 costs $O(q T_{\mathsf{MMX}}(k))$ operations. Step 3 requires $O(m(qk)^{\omega-1})$ operations. Step 4 executes $O(T_{\mathsf{MMX}}(qk) + m(qk)^{\omega-1})$ operations. The result is the $qk \times qk$ matrix $\widetilde{\mathbf{M}}$ on which we call COUNT on each iteration. From Lemma E.1, assuming that we have regularized $\widetilde{\mathbf{M}}$ appropriately using $\mathsf{REGULARIZE}$, COUNT costs at most $O((qk)^\omega \operatorname{polylog}(qk/\epsilon'_m)) = O((qk)^\omega \operatorname{polylog}(qk/\operatorname{gap}_k(\mathbf{M})))$. The maximum number of bits is in the last iteration, which is equal to $O\left(\operatorname{polylog}\left(\frac{mq\kappa(\mathbf{K})}{\operatorname{gap}_k(\mathbf{M})}\right)\right)$.

Step 6: Principal singular vectors. Given a suitable approximation for the gap and the midpoint, we next use Lemma B.1 to prove forward error bounds between the true k-spectral projector of $\mathbf{M} = \mathbf{\Phi}^\top \mathbf{Q}^\top \mathbf{X} \mathbf{X}^\top \mathbf{Q} \mathbf{\Phi}$ and the one of $\widetilde{\mathbf{M}} = \mathbf{M} + \mathbf{E_M}$. From Lemma B.1, if we set $\mu = \widetilde{\mu}_k$, we have that $\|\operatorname{sgn}(\mathbf{M} - \widetilde{\mu}_k) - \operatorname{sgn}(\widetilde{\mathbf{M}} - \widetilde{\mu}_k)\| \le \epsilon_{\mathsf{SGN}}$ if $\|\mathbf{E_M}\| \le \epsilon_{\mathsf{SGN}} \frac{|\lambda_{\min}(\widetilde{\mu}_k - \mathbf{M})|^2 \pi}{128} = \epsilon_{\mathsf{SGN}} \Theta(\operatorname{gap}_k(\mathbf{M})^2)$. The same holds for the spectral projectors, i.e. $\|\mathbf{\Pi}_k(\mathbf{M}) - \mathbf{\Pi}_k(\widetilde{\mathbf{M}})\| \le \epsilon_{\mathsf{SGN}}$. The bound for $\|\mathbf{E_M}\|$ is achieved by setting $\mathbf{u} \le \frac{\epsilon_{\mathsf{SGN}} \widetilde{\operatorname{gap}}_k^2}{\operatorname{poly}(mq)\widetilde{\kappa}}$, which translates to $O\left(\log(\frac{mq\kappa(\mathbf{K})}{\epsilon_{\mathsf{SGN}}\widetilde{\operatorname{gap}}_k})\right)$ bits.

It remains to approximate $\mathbf{\Pi}_k(\widetilde{\mathbf{M}})$, denoted as $\mathbf{\Pi}_k$ for simplicity, and then use deflation, similar to the vanilla PCA Algorithm 7. We first set $\epsilon_{\mathsf{PURIFY}} = (\frac{\epsilon_{\mathsf{PCA}}^2}{20^3(qk)^4})^2$, and then use Algorithm 3,

$$\widetilde{\mathbf{\Pi}}_k \leftarrow \mathsf{PURIFY}(\widetilde{\mathbf{M}}, \mathbf{I}, \widetilde{\mu}_k, \widetilde{\operatorname{gap}}_k, \epsilon_{\mathsf{PURIFY}}),$$

followed by $\widetilde{\mathbf{U}}_k \leftarrow \mathsf{DEFLATE}(\widetilde{\mathbf{\Pi}}_k, k, \epsilon_{\mathsf{PURIFY}}, \epsilon_{\mathsf{PCA}})$. From Proposition 2.1, PURIFY costs $O((qk)^\omega \log(qk/\operatorname{gap}_k))$ floating point operations using

$$O(\operatorname{polylog}(\frac{qk}{\epsilon_{\mathsf{PURIFY}} \operatorname{gap}_k})) = O(\operatorname{polylog}(\frac{qk}{\epsilon_{\mathsf{PCA}} \operatorname{gap}_k}))$$

bits. From Theorem A.5, $\mathsf{DEFLATE}$ costs $O((qk)^\omega)$ operations using $O(\operatorname{polylog}(qk/\epsilon_{\mathsf{PCA}}))$ bits, and succeeds with high probability. The returned matrix $\widetilde{\mathbf{U}}_k$ satisfies $\|\widetilde{\mathbf{U}}_k - \mathbf{U}_k\| \le \epsilon_{\mathsf{PCA}}$, where $\mathbf{U}_k$ is a matrix whose columns form an orthonormal basis for the span of the top-k singular vectors of $\widetilde{\mathbf{M}}$. This also implies that $\|\widetilde{\mathbf{U}}_k \widetilde{\mathbf{U}}_k^\top - \mathbf{U}_k \mathbf{U}_k^\top\| \le 3\epsilon_{\mathsf{PCA}}$. Then

$$\|\mathbf{\Pi}_k(\mathbf{M}) - \widetilde{\mathbf{U}}_k \widetilde{\mathbf{U}}_k^\top\| \le \|\mathbf{\Pi}_k(\mathbf{M}) - \mathbf{\Pi}_k(\widetilde{\mathbf{M}})\| + \|\mathbf{\Pi}_k(\widetilde{\mathbf{M}}) - \widetilde{\mathbf{U}}_k \widetilde{\mathbf{U}}_k^\top\| \le \epsilon_{\mathsf{SGN}} + 3\epsilon_{\mathsf{PCA}}.$$

We finally return $\widetilde{\mathbf{Z}}_k = \mathsf{MM}(\widetilde{\mathbf{Q}}, \widetilde{\mathbf{U}}_k)$. It holds that where $\|\widetilde{\mathbf{Z}}_k - \widetilde{\mathbf{Q}}\widetilde{\mathbf{U}}_k\| \le \mathbf{u} \cdot O(\operatorname{poly}(mq))$. We can then write

$$\begin{aligned}
\widetilde{\mathbf{Z}}_k \widetilde{\mathbf{Z}}_k^\top &= \widetilde{\mathbf{Q}} \widetilde{\mathbf{U}}_k \widetilde{\mathbf{U}}_k^\top \widetilde{\mathbf{Q}}^\top + \mathbf{E_Z} \\
&= \widetilde{\mathbf{Q}}(\mathbf{\Pi}_k(\mathbf{M}) + \mathbf{E_\Pi})\widetilde{\mathbf{Q}}^\top + \mathbf{E_Z} \\
&= \widetilde{\mathbf{Q}}\mathbf{\Pi}_k(\mathbf{M})\widetilde{\mathbf{Q}}^\top + \widetilde{\mathbf{Q}}\mathbf{E_\Pi}\widetilde{\mathbf{Q}}^\top + \mathbf{E_Z} \\
&= (\mathbf{Q}\mathbf{\Phi} + \mathbf{E_Q})\mathbf{\Pi}_k(\mathbf{M})(\mathbf{Q}\mathbf{\Phi} + \mathbf{E_Q})^\top + \widetilde{\mathbf{Q}}\mathbf{E_\Pi}\widetilde{\mathbf{Q}}^\top + \mathbf{E_Z} \\
&= \mathbf{Q}\mathbf{\Phi}\mathbf{\Pi}_k(\mathbf{M})\mathbf{\Phi}^\top \mathbf{Q}^\top + \mathbf{E}' + \widetilde{\mathbf{Q}}\mathbf{E_\Pi}\widetilde{\mathbf{Q}}^\top + \mathbf{E_Z},
\end{aligned}$$

where $\mathbf{Q}\mathbf{\Phi}\mathbf{\Pi}_k(\mathbf{M})\mathbf{\Phi}^\top \mathbf{Q}^\top = \mathbf{Z}_k \mathbf{Z}_k^\top$ is equivalent to the true, exact arithmetic $\mathbf{Z}_k$ of Algorithm 8. The error matrices satisfy $\|\mathbf{E_Z}\| \le \mathbf{u} \cdot O(\operatorname{poly}(mq))$, $\|\widetilde{\mathbf{Q}}\mathbf{E_\Pi}\widetilde{\mathbf{Q}}^\top\| \le O(\epsilon_{\mathsf{SGN}} + \epsilon_{\mathsf{PCA}})$, and $\|\mathbf{E}'\| \le O(\|\mathbf{E_Q}\|) \in O(\|\mathbf{E_M}\|) \in O(\epsilon_{\mathsf{SGN}} \operatorname{gap}_k(\mathbf{M})^2)$.

Putting everything together. We can now state the main result by summarizing the Steps 1-6.

Theorem G.2 (Restatement of Theorem 4.2). *Let $\mathbf{X}$ be a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\|\mathbf{X}\| \le 1$, $k \in [n]$ a target rank, $\epsilon_{\mathsf{PCA}} \in (0, 1)$ an accuracy parameter, and $q = \Theta\left(\frac{\log(n)}{\sqrt{\epsilon_{\mathsf{PCA}}}}\right)$. Let $T_{\mathsf{MMX}}(k)$ denote the*

67

complexity to stably multiply $\mathbf{X}$ *or* $\mathbf{X}^\top$ *with a dense matrix with* k *columns from the right (see Def. G.1). Using the Steps 1-6 that are detailed in Appendix G.3 as a floating point implementation of Algorithm 8, we can compute a matrix* $\widetilde{\mathbf{Z}}_k \in \mathbb{R}^{m \times k}$ *that satisfies*

$$\left\| \widetilde{\mathbf{Z}}_k \widetilde{\mathbf{Z}}_k^\top - \mathbf{Z}_k \mathbf{Z}_k^\top \right\| \leq O(\epsilon_{\mathsf{PCA}}),$$

with high probability, where $\mathbf{Z}_k$ *is an approximate basis for the top-k principal components of* $\mathbf{X}$, *returned by Algorithm 8 in exact arithmetic. The total cost is at most*

$$O\left(q T_{\mathsf{MMX}}(k) \log(\tfrac{\kappa(\mathbf{K})}{\mathrm{gap}_k(\mathbf{M})}) + m(qk)^{\omega-1} \log(\tfrac{1}{\mathrm{gap}_k(\mathbf{M})}) + (qk)^\omega \operatorname{polylog}(\tfrac{qk}{\mathrm{gap}_k(\mathbf{M})}) \right)$$

floating point operations, using $O\left(\operatorname{polylog}(\tfrac{mq\kappa(\mathbf{K})}{\epsilon_{\mathsf{PCA}} \, \mathrm{gap}_k}) \right)$ *bits of precision.* $\mathbf{K}, \mathbf{M}$ *are the same as in Alg. 8. The only parameters that are initially required are* k *and* ϵ_{PCA}.

Proof. We first compute $\widetilde{\kappa} \in \Theta(\kappa(\mathbf{K}))$ in Steps 1 and 2. This costs

$$O\left(q \log(\kappa(\mathbf{K})) T_{\mathsf{MMX}}(k) \right)$$

floating point operations using $O(\operatorname{polylog}(qm\kappa(\mathbf{K})))$ bits.

Then, in Step 5 we iteratively use Steps 1, 3, and 4 to compute the midpoint and the gap, in a total of $O(\log(1/\mathrm{gap}_k))$ iterations. The total cost is

$$O\left(\left(q T_{\mathsf{MMX}}(k) + m(qk)^{\omega-1}\right) \log(\tfrac{1}{\mathrm{gap}_k(\mathbf{M})}) + (qk)^\omega \operatorname{polylog}(\tfrac{qk}{\mathrm{gap}_k(\mathbf{M})}) \right)$$

floating point operations using at most

$$O\left(\operatorname{polylog}(\tfrac{mq\kappa(\mathbf{K})}{\mathrm{gap}_k(\mathbf{M})}) \right)$$

bits of precision. The only parameter that we require to know beforehand, except for the matrix sizes, is $\widetilde{\kappa}$ from the previous step.

In the last Step 6, assuming that $\epsilon_{\mathsf{SGN}} = \epsilon_{\mathsf{PCA}}$, we need to run Steps 1, 3, and 4, using $O\left(\operatorname{polylog}(\tfrac{mq\kappa(\mathbf{K})}{\epsilon_{\mathsf{PCA}} \, \mathrm{gap}_k}) \right)$ bits. The number of arithmetic operations does not exceed the one from the previous step. Thereafter the costs of DEFLATE, PURIFY, and MM are negligible compared to the costs of Steps 1, 3, and 4. The final matrix $\widetilde{\mathbf{Z}}_k$ satisfies

$$\left\| \widetilde{\mathbf{Z}}_k \widetilde{\mathbf{Z}}_k^\top - \mathbf{Z}_k \mathbf{Z}_k^\top \right\| \leq O(\epsilon_{\mathsf{PCA}}),$$

where $\mathbf{Z}_k$ is the true, exact arithmetic projector in Algorithm 8.

Summarizing everything, we can compute $\widetilde{\mathbf{Z}}_k$ as advertised above in a total of at most

$$O\left(q T_{\mathsf{MMX}}(k) \log(\tfrac{\kappa(\mathbf{K})}{\mathrm{gap}_k(\mathbf{M})}) + m(qk)^{\omega-1} \log(\tfrac{1}{\mathrm{gap}_k(\mathbf{M})}) + (qk)^\omega \operatorname{polylog}(\tfrac{qk}{\mathrm{gap}_k(\mathbf{M})}) \right)$$

floating point operations, using $O\left(\operatorname{polylog}(\tfrac{mq\kappa(\mathbf{K})}{\epsilon_{\mathsf{PCA}} \, \mathrm{gap}_k}) \right)$ bits of precision. $\qquad\square$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims accurately follow the paper contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations have been clearly discussed in the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theoretical results include an informal proof in the main paper and a reference to the full, rigorous proof in the appendix. All definitions and assumptions are clearly stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: There are no experiments, only theoretical results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: There are no experiments and no code associated with the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: There are no experiments in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:
   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no apparant societal impact of the work performed as it only includes mathematical results.

    Guidelines:
    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: There is no data or models involved in the paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.