Randomized Sketching for Generalized Eigenvalue Problems

Surya Appana

Advisors: Ryan Schneider and James Demmel

University of California, Berkeley

Fall 2025 – Spring 2026

Abstract

This work investigates how random matrix sketching can accelerate algorithms for generalized eigenvalue problems (GEPs). We further explore how sketched Rayleigh-Ritz and sketched-QZ can be set up for GEPs and compare their performance (speed and residual error) in well-conditioned and ill-conditioned settings. The results show that randomized sketches can significantly reduce runtime while maintaining high eigenvalue accuracy, highlighting the promise of randomized algorithms for (non-symmetric) GEPs in addition to standard eigenvalue problems. The complete code can be found at `github.com/SurCalGit/sketching_eigenvalue_problems.git`.

Keywords: Generalized eigenvalue problems (GEPs), Krylov Subspace Methods, Sketch-and- solve paradigm, Rayleigh-Ritz, Subspace embedding

1 Background and Motivation

1.1 Generalized eigenvalue problems

In the standard nonsymmetric eigenvalue problem (EP), given an n-by-n matrix A, we aim to find (λ, x) such that

$$Ax = \lambda x.$$

This problem is already ubiquitous in numerical linear algebra and in a variety of applications, including machine learning [1, 3]. In the generalized eigenvalue problem (GEP), we are given a matrix pair (A, B) (with both matrices of the same size) and aim to find (λ, x) such that

$$Ax = \lambda Bx.$$

In cases where B is invertible, this can be turned into an eigenvalue problem $B^{-1}Ax = \lambda x$, although B may be poorly conditioned and inverting may be computationally expensive. We call $A - \lambda B$ as a function of λ a *matrix pencil*. In cases where $\det(A - \lambda B) = 0$ for all λ, the matrix pair is called *singular*. Otherwise, it is *regular*. We are typically interested in the regular case, in which case λ and x satisfying $Ax = \lambda Bx$ are called eigenvalues and eigenvectors respectively for the pencil (A, B). The following table shows analogies between EPs and GEPs (see [9, Section 6] for further background). Assume that (A, B) of order n is **regular** unless otherwise stated.

Concept	$Ax = \lambda x$	$Ax = \lambda Bx$
Eigenvalues	A has n complex and finite eigenvalues which solve $\det(A - \lambda I) = 0$	(A, B) has n (generalized) eigenvalues. We have infinite eigenvalues when B is singular.
Rayleigh Quotient	$R(A, x) = \frac{x^T Ax}{x^T x}$	$R(A, B, x) = \frac{x^T Ax}{x^T Bx}$
Similarity/Equivalence	Matrices A, B are similar if there exists nonsingular S such that $B = S^{-1}AS$. Similar matrices have the same eigenvalues.	If X and Y are nonsingular, we call the pairs (A, B) and (YAX, YBX) equivalent.
Schur Decomposition	There exists unitary U such that $U^H AU = T$ is upper triangular. The eigenvalues of A appear on the diagonal of T.	There exist unitary U, V such that matrices in the equivalent pair $(S, T) = (V^H AU, V^H BU)$ are upper-triangular.
Jordan/Weierstrass Canonical form	For A, there exists nonsingular X such that $X^{-1}AX$ is block diagonal with Jordan blocks $J_{k_i}(\lambda_i) \in \mathbb{C}^{k_i \times k_i}$.	(A, B) is equivalent to a form $[(J, I), (I, N)]$, where J and N are in Jordan canonical form and N is nilpotent.
Structured Problems (with real eigenvalues)	A is Hermitian if $A = A^H$. A has real eigenvalues and a unitary diagonalization.	If (A, B) is Hermitian (A and B are Hermitian), it does not necessarily have real (generalized) eigenvalues. The Hermitian pair is a *definite pair* if the quantity $\gamma(A, B) := \min_{x \in \mathbb{C}^n, \|x\|_2 = 1} |x^H (A + iB)x|$ is strictly positive. A definite pair (A, B) is regular, and there exists nonsingular X such that $X^H AX$ and $X^H BX$ are diagonal. All definite pairs have real eigenvalues.

Table 1: Comparison between standard and regular generalized eigenvalue problems

1.2 Direct Methods: QZ Algorithm

The QZ Algorithm [6] is the classical direct method for obtaining a full set of eigenvalues for a matrix pair by computing the generalized Schur decomposition of the (regular) pair. The primary goal of the QZ algorithm is to find unitary matrices Q and Z that simultaneously transform A and B into upper triangular forms.

Eigenvalues

The generalized eigenvalues of the pencil (A, B) can be directly read from the diagonals of the resulting triangular matrices:

- If $T_{B_{ii}} \neq 0$, the eigenvalues are $\lambda_i = \frac{T_{A_{ii}}}{T_{B_{ii}}}$.
- If for some k, $T_{A_{kk}} = T_{B_{kk}} = 0$, then the spectrum $\sigma(A, B) = \mathbb{C}$ and the pencil is singular.

Algorithmic Steps

The algorithm generally proceeds in the following stages:

1. **Initial Reduction:** matrices A and B are transformed into Hessenberg and upper-triangular forms respectively.
2. **Deflation:** Zeros are deflated in the lower sub-diagonal of the Hessenberg matrix and on the diagonal of the upper triangular matrix.
3. **Iterative Phase:** An implicit shift QR algorithm is applied to the product AB^{-1} (implicitly, without forming B^{-1}) to iteratively reduce the Hessenberg matrix to triangular form while maintaining the triangularity of B.

Runtime and Complexity

For a full $n \times n$ matrix pair, the algorithm is a direct-style iterative method with a runtime generally scaling as $O(n^3)$ with space complexity $O(n^2)$. Thus, even if we would like to compute a small subset of eigenvalues, the overall runtime scales cubicly with the problem size n since the full QZ reduction must still be performed.

1.3 Krylov subspace methods

Algorithm 1 The Arnoldi algorithm for (partial) reduction to Hessenberg form

Input: $A \in \mathbb{C}^{n\times n}$, initial vector $b \in \mathbb{C}^n$, Krylov basis dimension d.
Output: $Q \in \mathbb{R}^{n\times n}$, $H \in \mathbb{C}^{d\times d}$

$q_1 \leftarrow b/\|b\|_2$ ▷ d is the number of columns of Q and H to compute
for $j = 1$ **to** d **do**
  $z \leftarrow Aq_j$
  for $i = 1$ **to** j **do**
    $h_{i,j} \leftarrow q_i^T z$
    $z \leftarrow z - h_{i,j} q_i$
  end for
  $h_{j+1,j} \leftarrow \|z\|_2$
  if $h_{j+1,j} = 0$ **then**
    quit
  end if
  $q_{j+1} \leftarrow z/h_{j+1,j}$
end for

For large applications, especially when the matrix A is only accessible through matrix-vector multiplications $x \mapsto Ax$, finding all eigenpairs is computationally intensive due to the $O(n^2)$ cost of storing A and the $O(n^3)$ cost of calculations, motivating iterative methods over direct methods such as the iterative QR algorithm. Central among these are **Krylov Subspace Methods (KSMs)**, which seek to find approximate eigenvectors located in a Krylov subspace [3]:

$$K_p(A, b) := \text{span}\{b, Ab, A^2b, \ldots, A^{p-1}b\}$$

This subspace often contains an excellent approximation to an eigenvector, even for a modest depth p, since it effectively takes the first few vectors from the power method. Suppose b is random and $K := [b, Ab, \ldots, A^{n-1}b]$ is invertible. Since $K^{-1}AK =: C$ is upper Hessenberg, we can approximate the eigenvalues of A with the eigenvalues of the simpler matrix C.

However, K is likely to be highly ill-conditioned since, by construction, $A^k b$ for $k = 0, 1, 2, \ldots$ approaches the eigenvector of A associated with the largest eigenvalue. Let $K = QR$ be the QR decomposition of K. Then $Q^T AQ = RCR^{-1} =: H$, which is also upper Hessenberg. Thus, if we can find an orthogonal matrix Q whose columns span the column space of K, we can determine approximate eigenvalues of A. In practice,

we only care about computing the first d columns of Q and approximately determining d eigenvalues of A (where $d \ll n$). Let $Q = [Q_d, Q_u]$ where Q_d is the first d columns of Q and Q_u contains the rest:

$$H = Q^T AQ = \begin{bmatrix} Q_d^T AQ_d & Q_d^T AQ_u \\ Q_u^T AQ_d & Q_u^T AQ_u \end{bmatrix} \equiv \begin{bmatrix} H_d & H^{ud} \\ H_{du} & H_u \end{bmatrix}$$

Note that H_d is upper Hessenberg and H_{du} contains a single, possibly nonzero entry at its upper right corner. If this is zero or close to zero, then the eigenvalues of H_d are exactly or approximately equal to eigenvalues of A. To compute the first d columns of Q, the classical approach is the *Arnoldi process* as shown in Algorithm 1.

Algorithm 2 k-truncated Arnoldi algorithm

Input: $A \in \mathbb{C}^{n\times n}$, initial vector $b \in \mathbb{C}^n$, Krylov basis dimension d, k for partial orthogonalization.
Output: $Q \in \mathbb{R}^{n\times n}$, $H \in \mathbb{C}^{d\times d}$

$q_1 \leftarrow b/\|b\|_2$ ▷ d is the total number of iterations, k is the truncation window
for $j = 1$ **to** d **do**
  $z \leftarrow Aq_j$
  $i_{\min} \leftarrow \max(1, j - k + 1)$
  for $i = i_{\min}$ **to** j **do**
    $h_{i,j} \leftarrow q_i^T z$
    $z \leftarrow z - h_{i,j} q_i$
  end for
  $h_{j+1,j} \leftarrow \|z\|_2$
  if $h_{j+1,j} = 0$ **then**
    quit
  end if
  $q_{j+1} \leftarrow z/h_{j+1,j}$
end for

This process requires $O(n^2 d)$ arithmetic operations for d iterations. Once we have obtained an orthonormal basis for $K_d(A, b)$ using the Arnoldi Process, we can then approximate the eigenvalues of A with the eigenvalues of H_d and multiply the associated eigenvectors by Q to obtain approximate eigenvectors for A. This approach is known as the *Rayleigh-Ritz* method, with approximate eigenvalues and eigenvectors known as the *Ritz values* and *Ritz vectors*. In practice, we often perform the *k-truncated Arnoldi process*, in which each new vector is orthogonalized against only the most recent k vectors, leading to an additional cost (over the generation of the vectors which is still $O(n^2 d)$) of $O(ndk)$ instead of $O(nd^2)$. See Algorithm 2.

For the generalized eigenvalue problem, given a matrix pair (A, B), we can compute a Kyrlov basis using either the Arnoldi or k-truncated Arnoldi process for $B^{-1}A$ without explicitly forming B^{-1}. After obtaining the basis Q_d, we formulate the projected problem $Q_d^T AQ_d x = \lambda Q_d^T BQ_d x$ and solve via the QZ algorithm. This is shown in Algorithm 3.

Algorithm 3 k-truncated Arnoldi for $Ax = \lambda Bx$ + Rayleigh–Ritz

Input: $A, B \in \mathbb{C}^{n\times n}$, initial vector $b \in \mathbb{C}^n$, Krylov basis dimension d, k for partial orthogonalization, residual tolerance τ.

Output: Approximate eigenpairs (λ_i, x_i) such that $Ax_i \approx \lambda_i Bx_i$.

$q_1 \leftarrow b/\|b\|_2$ ▷ d = number of iterations, k = truncation window
for $j = 1$ **to** d **do**
  $w \leftarrow Aq_j$
  Solve $Bz = w$
  $i_{\min} \leftarrow \max(1, j - k + 1)$
  for $i = i_{\min}$ **to** j **do**
    $h_{i,j} \leftarrow q_i^T z$
    $z \leftarrow z - h_{i,j} q_i$
  end for
  $h_{j+1,j} \leftarrow \|z\|_2$
  if $h_{j+1,j} = 0$ **then**
    quit
  end if
  $q_{j+1} \leftarrow z/h_{j+1,j}$
end for
$Q_d \leftarrow [q_1, \ldots, q_d]$
$H_d \leftarrow Q_d^T A Q_d$
$G_d \leftarrow Q_d^T B Q_d$
Solve via QZ Algorithm: $H_d y = \lambda G_d y$
Return approximate eigenpairs (λ, x) **for each** y**:** λ, $x \leftarrow Q_d y$

If the linear solve with B is fast ($O(n^2)$), performing k-truncated Arnoldi and solving the reduced generalized problem has overall time complexity $O(n^2 d + ndk + nd^2 + d^3)$, accounting for linear solves with B, the k-truncated Arnoldi process, forming the reduced problem $O(nd^2)$, and solving the reduced problem $O(d^3)$. The dominant cost is basis generation, within which the dominant cost is matrix-vector multiplication $x \mapsto Ax$ ($O(n^2)$). However, if A is structured or sparse, performing this multiplication can be done faster, yielding a general cost expression for basis generation as $O(d\times\text{MatVecCost})$ where **MatVecCost** is the cost of matrix-vector multiplication.

1.4 Randomized matrix sketching

Randomized embeddings are motivated by the need to develop fast, reliable, and scalable algorithms in numerical linear algebra by reducing dimensionality. One important desirable quality of a random embedding on an N-point random subset of $\mathbb{R}^n$ is to approximately preserve distances between points (within a small factor). The following well-known lemma shows that when $k = O(\log(N)/\epsilon^2)$, such an embedding exists [4].

Lemma (Johnson-Lindenstrauss)**.** Let N be any integer and $\epsilon \in (0, 1)$. Suppose k is a positive integer such that $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log(n)$. Then for any set X of N points in $\mathbb{R}^n$ there exists a map $f : \mathbb{R}^n \to \mathbb{R}^k$ such that for all $x, y \in X$, we have

$$(1 - \epsilon)\|x - y\|^2 \leq \|f(x) - f(y)\|^2 \leq (1 + \epsilon)\|x - y\|^2,$$

where this norm is the ℓ_2 norm.

In practice, we apply the "sketch-and-solve" paradigm [5, 8, 10] to reduce the dimension of a problem by multiplying a large matrix by a randomized *subspace embedding* matrix with a much smaller output dimension and solving the reduced problem. A subspace embedding is defined as follows.

Defn (Subspace Embedding)**.** Suppose $A \in \mathbb{C}^{n\times d}$ whose columns span the subspace $V \subseteq \mathbb{C}^n$. Then $S \in \mathbb{C}^{s\times n}$ is a *subspace embedding for V with distortion ϵ* if for all $x \in \mathbb{C}^d$, we have

$$(1 - \epsilon)\|Ax\|_2 \leq \|SAx\|_2 \leq (1 + \epsilon)\|Ax\|_2.$$

In other words, S approximately preserves the ℓ_2 norm of vectors in the range of A.

Since the subspace V is typically unknown, we must draw the subspace embedding S randomly to satisfy the above property with high probability. Suppose A above has full column rank (and therefore V has dimension d). To embed V with distortion ϵ, we optimally have $s \approx d/\epsilon^2$. There are a few popular ways of constructing a randomized subspace embedding that work for the eigenvalue problem.

1.4.1 Gaussian Sketches

A simple Gaussian Sketch that satisfies the subspace embedding property is as follows: let $S \in \mathbb{R}^{s\times n}$ have i.i.d. entries $\mathcal{N}(0, 1/s)$ and define $\widetilde{A} = SA$. If $s = O((d+\log(1/\delta))/\varepsilon^2)$, then with probability $1-\delta$ ($\delta \in (0,1)$)

$$(1-\varepsilon)\|x\|_2^2 \leq \|Sx\|_2^2 \leq (1+\varepsilon)\|x\|_2^2$$

for all x in any fixed d-dimensional subspace. Thus, S approximately preserves the geometry of range(A).

1.4.2 SRFTs (Subsampled random Fourier transforms)

These subspace embeddings take the form

$$S = \sqrt{\frac{n}{s}} DFE \in \mathbb{C}^{s\times n},$$

where $D \in \mathbb{C}^{s\times n}$ diagonally projects onto s independently chosen random coordinates, $F \in \mathbb{C}^{n\times n}$ is the unitary discrete Fourier transform (DFT), and $E \in \mathbb{C}^{n\times n}$ is a diagonal matrix whose entries are independent Steinhaus random variables (uniform on the complex unit circle). Using the subsampled FFT algorithm, we can apply S to an $n \times d$ matrix in $O(nd\log(d))$ operations.

1.4.3 Sparse Maps

A sparse random embedding takes the form

$$S = \frac{1}{\sqrt{s}}[s_1, \ldots, s_n] \in \mathbb{C}^{s\times n},$$

where each column s_i has a fixed number of nonzero entries ζ drawn from the Steinhaus distribution whose coordinates are uniformly random. This embedding is cheaper to apply on sparse data.

1.5 Randomized sketching for eigenvalue problems

Recognizing the computational limitations inherent in traditional KSMs, Nakatsukasa and Tropp [7] introduced the integration of random matrix sketching into subspace projection methods such as Rayleigh-Ritz (RR). This approach leverages the "sketch-and-solve paradigm" to reduce the dimensionality of the underlying computations. Suppose $V \in \mathbb{C}^{n\times d}$ is such that its column span contains approximate eigenvectors for A (e.g., computed through a truncated Arnoldi process). The classic Rayleigh-Ritz (RR) method solves the least-squares minimization problem:

$$\min_{M\in\mathbb{C}^{d\times d}} \|AV - VM\|_F.$$

The reduced EP is to find eigenvalues and eigenvectors for $M^* = V^\dagger AV$. By drawing a random sketching matrix $S \in \mathbb{C}^{s\times n}$ (the authors suggest $s = 4d$), the RR problem is transformed into the **sketched Rayleigh-Ritz (sRR)** problem:

$$\min_{M\in\mathbb{C}^{d\times d}} \|S(AV - VM)\|_F.$$

The resulting solution $\hat{M} = (SV)^\dagger(SAV)$ is computed significantly faster than the optimal solution M^*: computing M^* requires $O(nd^2)$ operations with the pseudoinverse computation as the bottleneck, whereas computing $\hat{M}$ requires $O(nd\log(d) + d^3)$ operations using an SRFT sketch (fast) with $s = O(d)$. This randomized approach gives immense flexibility in selecting the basis V. Specifically, highly conditioned bases,

where $\kappa_2(V) \lesssim u^{-1}$ (where u is the unit roundoff), can be tolerated, allowing the use of computationally cheaper, non-orthogonal basis generation techniques, such as k-truncated Arnoldi. Overall, the sketched approach yields substantial practical speedups, often exceeding $10\times$ faster than optimized standard routines in tested examples.

A variational formulation of RR is introduced again for the GEP. Suppose we want to solve $Ax = \lambda Bx$ for $A, B \in \mathbb{C}^{n\times n}$ and nonzero x. Further suppose that $V \in \mathbb{C}^{n\times d}$ contains approximate eigenvectors. Our goal is to solve:

$$\min_{M\in\mathbb{C}^{d\times d}} \|AV - BVM\|_F. \tag{1}$$

One solution is $M^* = (BV)^\dagger(AV)$, for which we can propose the standard EP. For the sketched problem, given a subspace embedding S for Range($[AV, BV]$), we have the following:

$$\min_{M\in\mathbb{C}^{d\times d}} \|S(AV - BVM)\|_F. \tag{2}$$

This has a solution $\hat{M} = (SBV)^\dagger(SAV)$, yielding a reduced EP. Once again, excluding the generation of basis V, we again have cost $O(d^3 + nd\log(d))$ compared to $O(nd^2)$ in classic RR. An alternative approach that does not require computing $(SBV)^\dagger$ is proposing the reduced GEP for the matrix pair (SAV, SBV), which is advantageous if SBV is ill-conditioned, potentially as a result of an ill-conditioned B. This reduced GEP can be solved by applying the QZ algorithm when $s = d$ or $s = O(d)$ by first padding SAV and SBV with zeros [1].

When the matrix B is symmetric positive definite but ill-conditioned, we can precondition the GEP with $B^{-1/2}$. Instead of explicitly forming $B^{-1/2}$, we work with triangular solves involving L and L^T where we take the Cholesky factorization $B = LL^T$ with L being lower triangular. This allows us to transform the generalized eigenvalue problem into $L^{-1}AL^{-T}y = \lambda y$, which can be accessed implicitly via solves with L and L^T. This approach avoids forming dense matrix square roots while retaining numerical stability and efficiency, since triangular solves can be performed in $O(n^2)$ time for dense matrices and faster for structured matrices.

1.6 Main Contributions

The following are the primary motivating questions in this work:

(1) Do the accuracy and runtime trends from Nakatsukasa and Tropp [7] extend to the nonsymmetric GEP and, in-particular, in the case of comparing eigenvalue distributions between the sketched and unsketched problems? How do sketched Rayleigh-Ritz and sketched QZ compare to the QZ algorithm on small problems (e.g. $n = 1000$) in terms of the distribution of computed individual eigenvalues?

(2) How does the performance of sketched algorithms compare between cases where B is well-conditioned and ill-conditioned? Does inverse-free sketched Rayleigh-Ritz (by applying QZ) provide more accurate eigenvalues compared to standard sketched Rayleigh-Ritz?

The six methods tested in this work are listed in Table 2. Algorithm 4 provides a detailed setup for sketched Rayleigh-Ritz for the GEP with options for inverse-free and preconditioned methods. Running this algorithm with these different options across small ($n = 1000$) and large ($n = 100,000$) problems, we observe a significant speed reduction in the sketched methods versus classical Rayleigh-Ritz while maintaining comparable eigenvalue accuracy as measured by both chordal distance and (normalized) residual error. Moreover, inverse-free sketched Rayleigh-Ritz does not appear to increase accuracy when B is ill-conditioned.

[1] Care must be taken in how this is set up to obtain meaningful eigenvalues.

Algorithm 4 k-truncated Arnoldi + Sketched Rayleigh-Ritz for GEP

Input: $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n\times n}$, $\mathbf{b} \in \mathbb{C}^n$, d, k, s (e.g., $s = 4d$), flags `precond`, `inversion_free`, `sketch`.
Output: Approximate eigenpairs $(\mathbf{x}_i, \lambda_i)$ for $\mathbf{Ax} = \lambda\mathbf{Bx}$.

if `precond` **then**
  Compute $\mathbf{B} = \mathbf{LL}^*$ or $\mathbf{B} = \mathbf{B}^{1/2}\mathbf{B}^{1/2}$.
  $\tilde{\mathbf{A}} \leftarrow \mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$, $\tilde{\mathbf{B}} \leftarrow \mathbf{I}$.
  $\mathbf{A} \leftarrow \tilde{\mathbf{A}}$, $\mathbf{B} \leftarrow \tilde{\mathbf{B}}$.
end if
if `precond` **then**
  Set $\mathbf{S} = V^T$.
else
  Draw $\mathbf{S} \in \mathbb{C}^{s\times n}$ (SRFT or sparse).
end if
Draw $\mathbf{S} \in \mathbb{C}^{s\times n}$ (SRFT or sparse).
$\mathbf{w}_1 = \mathbf{b}$, $\mathbf{b}_1 = \mathbf{w}_1/\|\mathbf{w}_1\|_2$, $\mathbf{m}_1 = \mathbf{A}\mathbf{b}_1$.
for $j = 2, \ldots, d$ **do**
  $\mathbf{w}_j = (\mathbf{I} - \mathbf{b}_{j-1}\mathbf{b}_{j-1}^* - \cdots - \mathbf{b}_{j-k}\mathbf{b}_{j-k}^*)\mathbf{m}_{j-1}$ ▷ Note: $\mathbf{b}_{-i} = 0$
  $\mathbf{b}_j = \mathbf{w}_j/\|\mathbf{w}_j\|_2$, $\mathbf{m}_j = \mathbf{A}\mathbf{b}_j$.
end for
$\mathbf{C} = \mathbf{S}[\mathbf{b}_1, \ldots, \mathbf{b}_d]$, $\mathbf{D} = \mathbf{S}[\mathbf{m}_1, \ldots, \mathbf{m}_d]$, $\mathbf{E} = \mathbf{S}[\mathbf{B}\mathbf{b}_1, \ldots, \mathbf{B}\mathbf{b}_d]$.
if `inversion_free` **then**
  Solve GEP via QZ: $\mathbf{D}\mathbf{y}_i = \lambda_i\mathbf{E}\mathbf{y}_i$.
else
  QR Factorization: $\mathbf{E} = \mathbf{UT}$.
  $\hat{\mathbf{M}} = \mathbf{T}^{-1}\mathbf{U}^*\mathbf{D}$.
  Solve standard EP: $\hat{\mathbf{M}}\mathbf{y}_i = \lambda_i\mathbf{y}_i$.
end if
$r_{\text{est},i} = \frac{\|\mathbf{D}\mathbf{y}_i - \lambda_i\mathbf{E}\mathbf{y}_i\|_2}{\|\mathbf{E}\mathbf{y}_i\|_2}$.
Identify $\mathcal{I}$ where $r_{\text{est},i} \le \tau$, set $\mathbf{x}_i = \mathbf{B}\mathbf{y}_i$.
if `precond` **then**
  $\mathbf{x}_i \leftarrow \mathbf{B}^{-1/2}\mathbf{x}_i$.
end if
return $(\mathbf{x}_i, \lambda_i)$ for $i \in \mathcal{I}$.

2 Test Problems

The six methods shown in Table 2 were tested on two applications of the nonsymmetric GEP: (1) exponential solutions to damped mass-spring systems (explained more in detail in [3]), and (2) finite-differences for solving a 1D elliptic eigenvalue problem. Furthermore, two cases were differentiated: well-conditioned B and ill-conditioned B.

2.1 Damped Mass-Spring Systems

Starting from a second-order system

$$A_1\ddot{u} + A_2\dot{u} + A_3u = 0,$$

where A_1 is a diagonal matrix representing masses, A_2 is a diagonal matrix representing damping coefficients, and A_3 is a tri-diagonal matrix that includes spring constants. Introduce the state vector $z = \begin{bmatrix} u \\ \dot{u} \end{bmatrix}$. This yields the first-order system

$$\begin{bmatrix} 0 & I \\ -A_3 & -A_2 \end{bmatrix} z = \lambda \begin{bmatrix} I & 0 \\ 0 & A_1 \end{bmatrix} z,$$

Method	**Inversion**	**Dominant Cost**	**Randomized**	**Runtime**
QZ algorithm for (A, B)	No	Dense factorization ($O(n^3)$)	No	$O(n^3)$
k-truncated Arnoldi + Rayleigh–Ritz (Algorithm 3)	Yes	Matrix-Vector Prod. ($O(nd^2)$)	No	$O(nd^2)$
k-truncated Arnoldi + QZ on (V^TAV, V^TBV) (Algorithm 4 with `sketch`=False)	No	Matrix-Vector Prod. + small QZ	No	$O(nd^2 + d^3)$
k-truncated Arnoldi + sketched Rayleigh–Ritz (SRFT) (Algorithm 4 with `sketch`=True)	Yes	Matrix-Vector Prod. + sketching	Yes	$O(nd \log d + d^3)$
k-truncated Arnoldi + sketched Inverse-free Rayleigh-Ritz (SRFT) (Algorithm 4 with `sketch`=True, `inverse_free`=True)	No	Matrix-Vector Prod. + sketching + small QZ	Yes	$O(nd \log d + d^3)$
k-truncated Arnoldi + sketched Inverse-free (SRFT) with (fast) Preconditioning (Algorithm 4 with `sketch`=True, `inverse_free`=True, `precond`=True)	No	Linear solves + sketching	Yes	$O(nd \log d)$

Table 2: Comparison of methods evaluated in this work, implemented with Algorithm 4. Runtime costs exclude the basis generation step ($O(n^2d + nkd)$)

which is a generalized eigenvalue problem. Typically, the mass matrix A_1 is well-conditioned, allowing efficient application of A_2^{-1} via direct inversion or linear solves. However, if A_1 is ill-conditioned, the GEP formulation above is preferable to prevent inversion.

2.2 1D Elliptic Eigenvalue Problem

This problem arises from the discretization of a one-dimensional elliptic differential operator, typically using finite difference or finite element methods. The resulting matrices exhibit a structured sparsity pattern (e.g., tridiagonal in the finite difference case), and the eigenvalues reflect the underlying differential operator spectrum.

$$-\frac{d}{dx}\left(\kappa(x)\frac{du}{dx}\right) = \lambda b(x)u(x),$$

with $x \in (0,1),\ u(0) = 0,\ u(1) = 0$. We discretize using a uniform grid $x_i = ih$, $h = \frac{1}{n+1}$. Using midpoint fluxes,

$$q_{i+\frac{1}{2}} \approx \kappa_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{h}.$$

the divergence term $-\frac{d}{dx}(\kappa u')|_{x_i}$ becomes

$$\approx \frac{1}{h^2}\Big(-\kappa_{i+\frac{1}{2}} u_{i+1} + (\kappa_{i+\frac{1}{2}} + \kappa_{i-\frac{1}{2}})u_i - \kappa_{i-\frac{1}{2}} u_{i-1}\Big).$$

The right-hand side is approximated by midpoint quadrature:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} b(x)u(x)\,dx \approx b(x_i)u_i h.$$

This yields matrix entries

$$A_{ii} = \frac{\kappa_{i-\frac{1}{2}} + \kappa_{i+\frac{1}{2}}}{h^2},$$
$$A_{i,i\pm1} = -\frac{\kappa_{i\pm\frac{1}{2}}}{h^2},$$

$$B_{ii} = b(x_i)h, \quad B_{ij} = 0\ (i \neq j).$$

$$Ax = \lambda Bx, \quad x = (u_1, \ldots, u_n)^T.$$

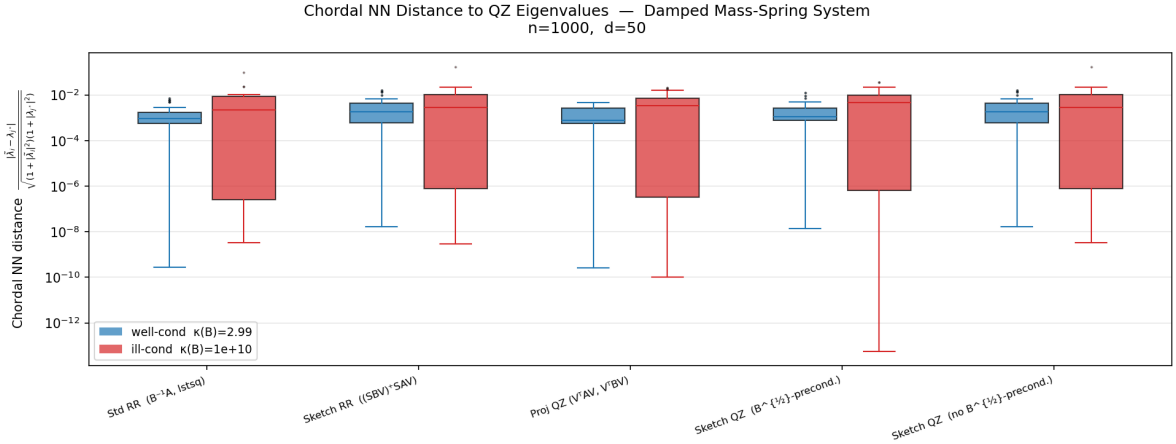
As shown above, the matrix A is tri-diagonal while B is diagonal, enabling fast inversion or linear solves.

3 Experimental Results

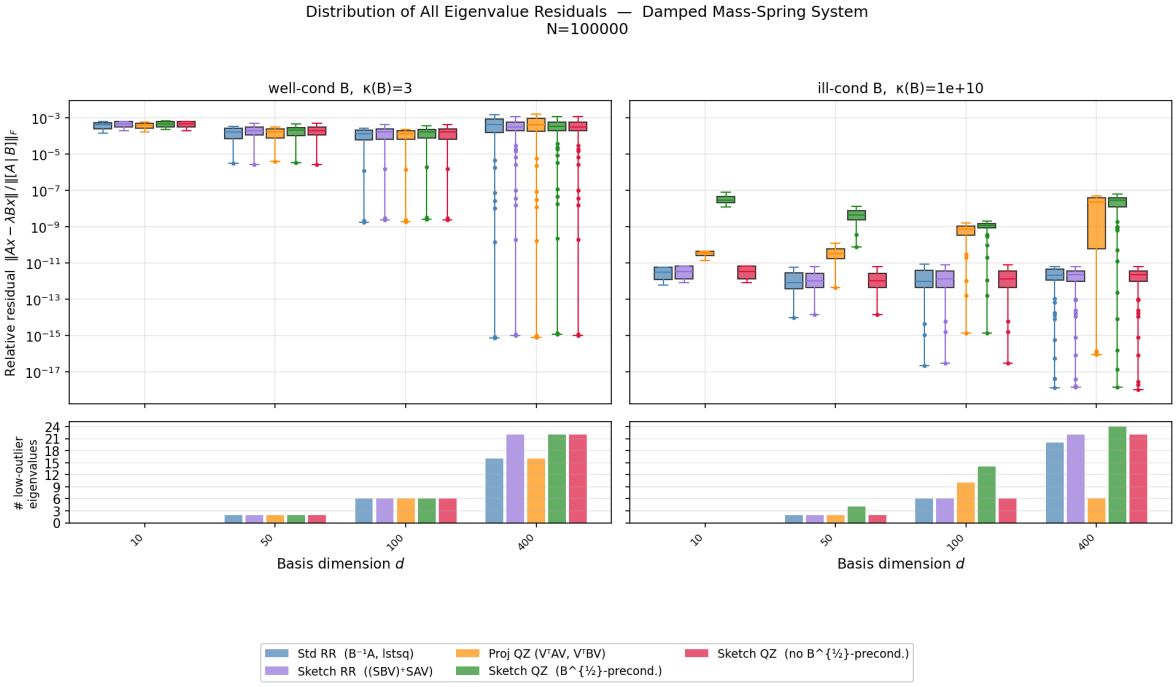
Table 2 shows the six methods tested, with the corresponding full pseudocode for the sketched methods displayed in Algorithm 4. Setting `precond=True` applies $B^{-1/2}$ preconditioning using a Cholesky factorization, and setting `inverse_free=True` applies the QZ algorithm to solve the reduced problem.

Accuracies of computed eigenvalues measured by chordal distance to QZ-eigenvalues (for $n = 1000$) and by residual error (for $n = 10^5$) are shown in figure 2 for the damped mass-spring system and elliptic eigenvalue problem respectively. In addition to the speedup observed by the sketched methods, we highlight the following observations:

- Sketched methods perform comparably to their unsketched counterparts across both applications.

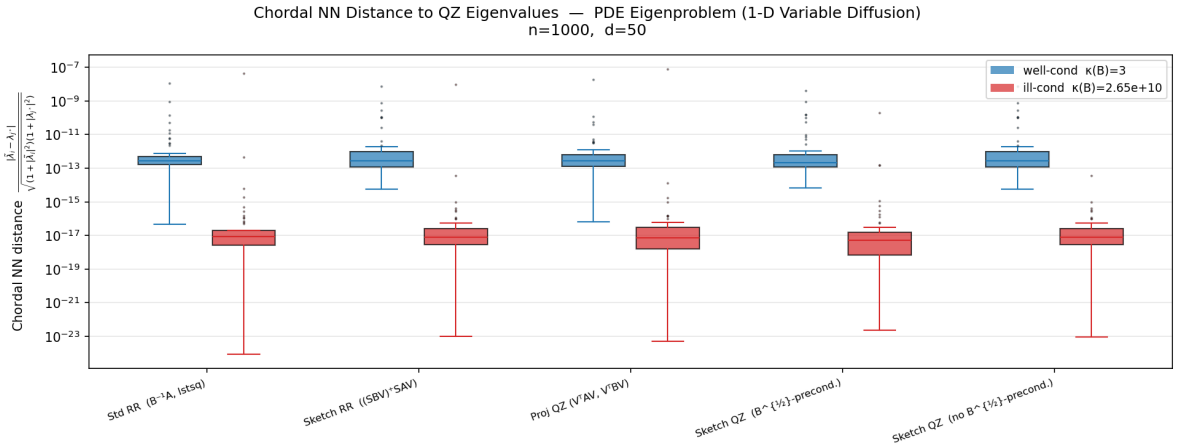


(a) Chordal nearest-neighbor distance between unsketched/sketched methods and full QZ eigenvalues for the damped mass-spring system ($n = 1000$, $d = 50$) across both well-conditioned and ill-conditioned B cases.

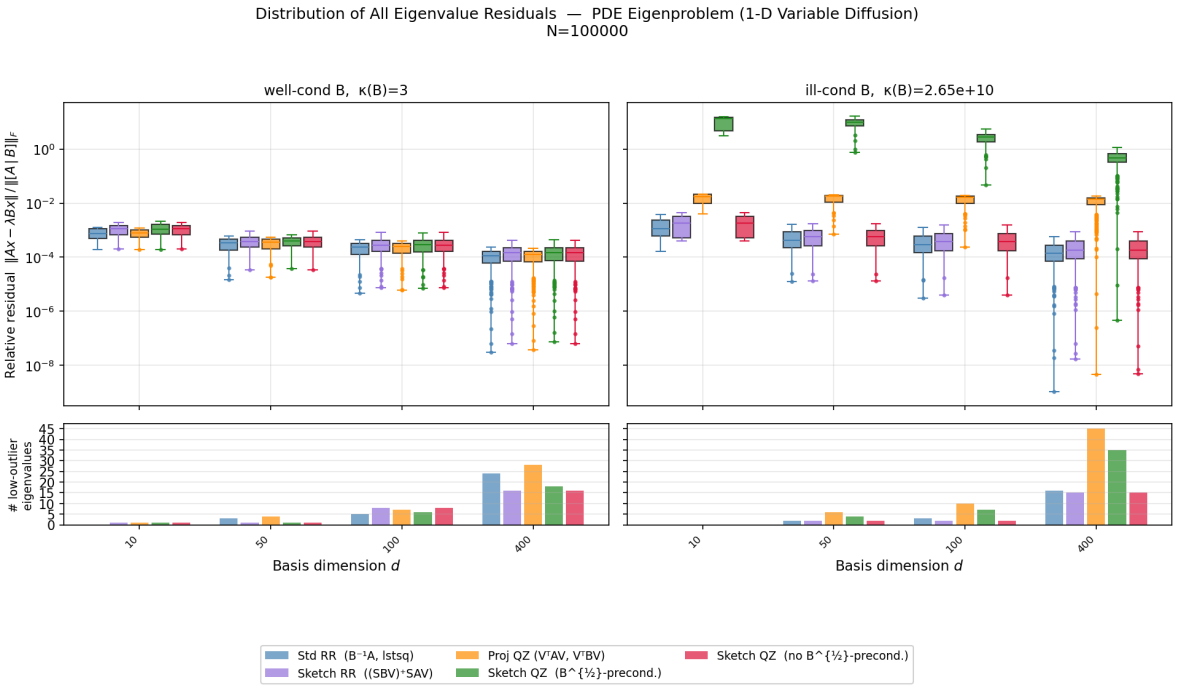


(b) Distribution of relative eigenpair residuals $|Ax - \lambda Bx|/(\|[A|B]\|_F)$ for the damped mass-spring system ($n = 10^5$). Top: residual distributions across basis dimension d. Bottom: number of low-residual outlier eigenvalues.

Figure 1: Accuracies of sketched methods on the damped mass-spring system measured by (a) chordal distances and (b) relative residual errors.



(a) Chordal nearest-neighbor distance to QZ eigenvalues for the 1D elliptic eigenvalue problem ($n = 1000$, $d = 50$) across both well-conditioned and ill-conditioned B cases.



(b) Distribution of relative residuals for the 1D elliptic eigenvalue problem ($N = 10^5$). Top: residual distributions versus basis dimension d. Bottom: count of low-residual outlier eigenvalues.

Figure 2: Accuracies of sketched methods on the 1D elliptic eigenvalue problem measured by (a) chordal distances and (b) relative residual errors.

- There is no substantial difference in accuracy between sketched QZ (without preconditioning) and sketched Rayleigh-Ritz.
- Preconditioning before QZ results in a drop in accuracy across both applications for the $n = 10^5$ case.

Experiments were designed using Python's SciPy library, using the `scipy.linalg.eig` and `scipy.linalg.qz` solvers to compute standard and generalized eigenvalues, and were executed on a machine with 80GB CPU RAM. The full code and experimental details for reproducibility can be found here: `github.com/SurCalGit/sketching_eigenvalue_problems.git`.

4 Discussion

The experimental results align closely with theoretical expectations from randomized numerical linear algebra and classical eigenvalue perturbation theory. The comparable performance between sketched and unsketched methods can be explained by the subspace preservation guarantees of randomized embeddings, as detailed in section 1.4. In the context of Rayleigh-Ritz, since the dominant computational cost of sketch-based methods lies in forming and decomposing large matrices, reducing the dimension from n to $s \ll n$ yields runtime savings (as shown in Figure 3) without significantly degrading accuracy (shown in Figure 2).

The observation that sketched QZ does not significantly outperform sketched Rayleigh-Ritz in the ill-conditioned B case may be understood by the fact that even when B is ill-conditioned, the generalized eigenvalues themselves may still be well-conditioned quantities, particularly if (A, B) satisfies favorable structural properties. Randomized sketching introduces its own approximation error, which can dominate any potential stability advantage gained by avoiding explicit inversion. As a result, both sketched Rayleigh-Ritz and sketched QZ exhibit similar accuracy in practice, even in regimes where B is poorly conditioned.

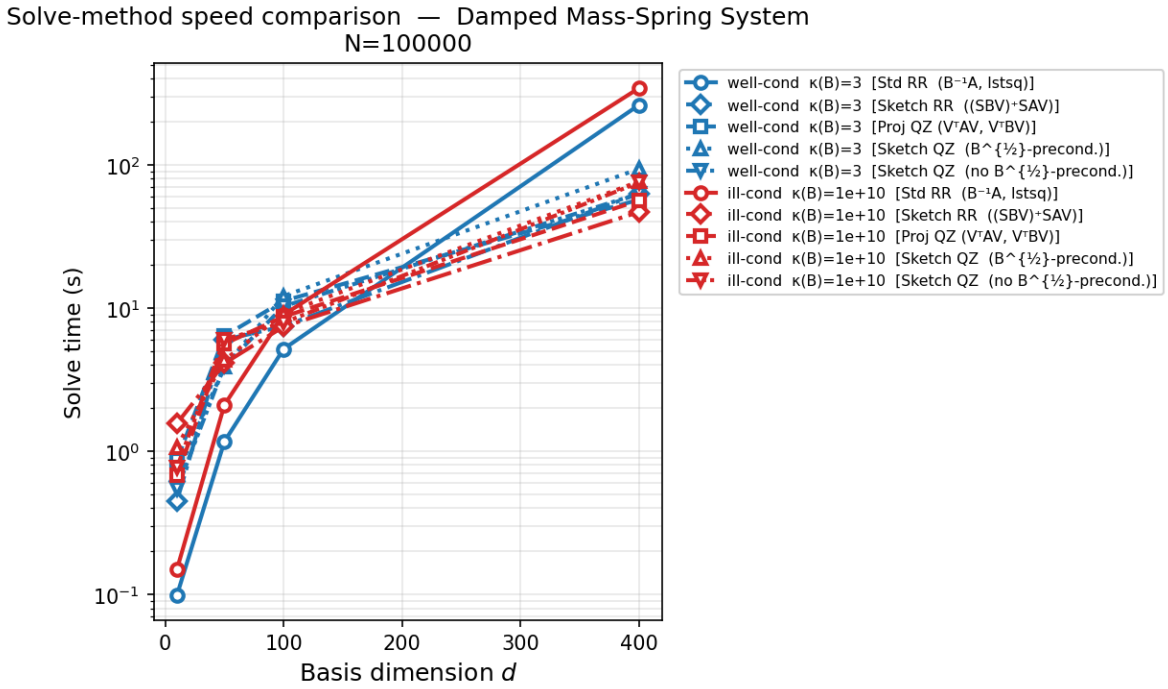
Furthermore, the elevated residual errors observed for the preconditioned method in the large-scale case ($n = 10^5$) are likely due to the numerical instability introduced by forming or applying $B^{1/2}$ or $B^{-1/2}$. Although preconditioning is appealing because it transforms the problem into a standard eigenvalue problem with improved conditioning, computing matrix square roots likely introduces approximation errors, which can accumulate and degrade the quality of the computed subspace. Additionally, applying these transformations may amplify floating-point errors when B has a wide spectrum, leading to loss of orthogonality and increased residuals in the final eigenpairs.

5 Conclusion

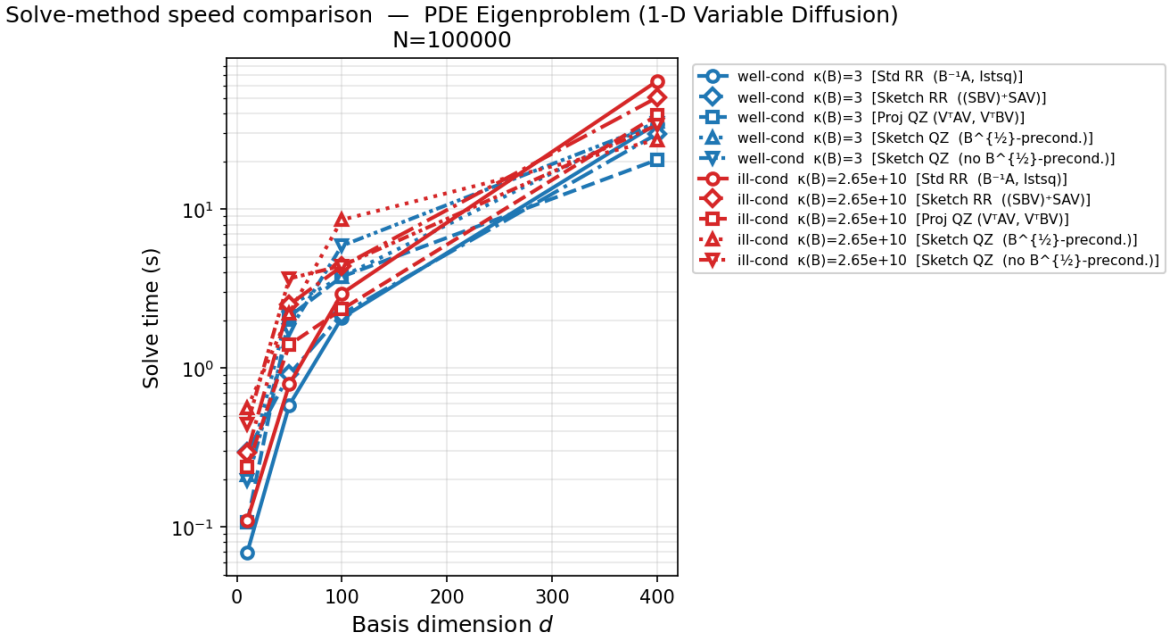
This work investigates the application of randomized sketching for faster solutions of (nonsymmetric) generalized eigenvalue problems (GEPs). While reducing runtime, sketched methods preserve accuracy of computed eigenvalues comparable to QZ and standard Rayleigh-Ritz (for both small and large problem sizes), where accuracy is measured in terms of both residual error and distance to "ground truth" eigenvalues computed by QZ.

Our experiments further indicate that sketched Rayleigh-Ritz and sketched QZ methods perform similarly in (terms of eigenvalue accuracy) across both cases where B is well-conditioned and ill-conditioned. Furthermore, while $B^{1/2}$ preconditioning can provide better eigenvalue solutions when B is ill-conditioned, in practice the performance is mixed and heavily depends on implementation details.

Overall, these results highlight the promise of randomized algorithms for solving large GEPs. Future work may focus on improving robustness under severe ill-conditioning, refining preconditioning strategies, and exploring alternative (potentially structured) sketches, such as those described in [2].



(a) Solve speed of sketched and unsketched methods for the damped mass-spring system ($n = 100000$, $d = 400$) across both well-conditioned and ill-conditioned B cases.



(b) Solve speed of sketched and unsketched methods for the 1D elliptic eigenvalue problem ($n = 100000$, $d = 400$) across both well-conditioned and ill-conditioned B cases.

Figure 3: Solve speeds (including basis generation) of sketched methods on the damped mass-spring system and elliptic eigenvalue problem.

References

[1] C.M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

[2] Chris Camaño, Ethan N. Epperly, Raphael A. Meyer, and Joel A. Tropp. Faster linear algebra algorithms with structured random matrices, 2025.

[3] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, January 1997.

[4] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

[5] Per-Gunnar Martinsson and Joel Tropp. Randomized numerical linear algebra: Foundations & algorithms, 2021.

[6] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis*, 10(2):241–256, 1973.

[7] Yuji Nakatsukasa and Joel A. Tropp. Fast and accurate randomized algorithms for linear systems and eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 45(2):1183–1214, 2024.

[8] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, page 143–152, USA, 2006. IEEE Computer Society.

[9] G. W. Stewart and Jiguang Sun. Matrix perturbation theory. 1990.

[10] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):1–157, October 2014.

A Computed Eigenvalue Distributions

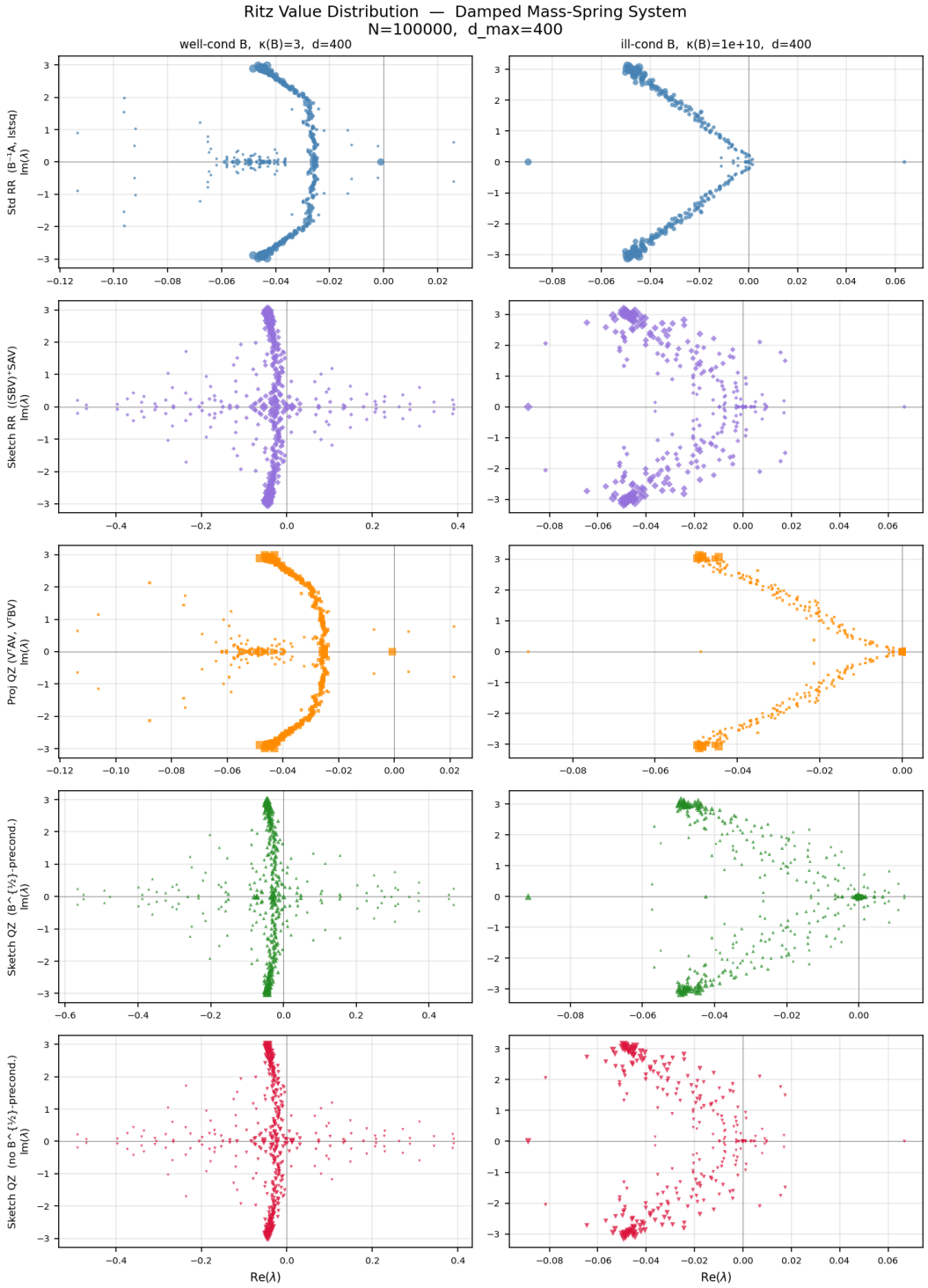


Figure 4: Ritz value distribution for mass-spring system eigenvalues: $n = 100000$, $d = 400$. Eigenvalues are displayed for well-conditioned and ill-conditioned B cases ($\kappa(B) = 3$ and $\kappa(B) = 10^{10}$ respectively).

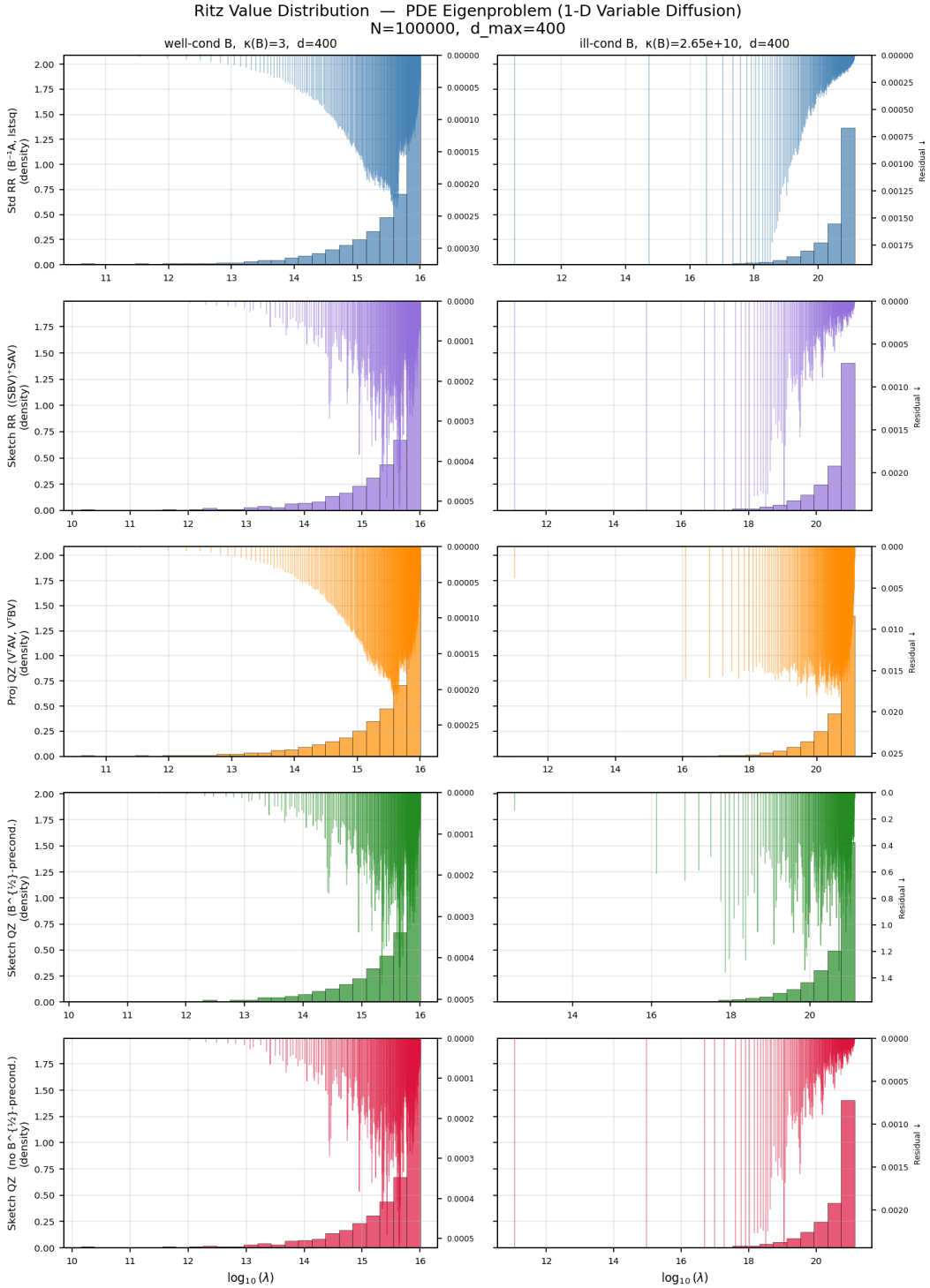


Figure 5: Ritz value distribution for 1D Elliptic Eigenvalue Problem: $n = 100000$, $d = 400$. Eigenvalues are displayed for well-conditioned and ill-conditioned B cases ($\kappa(B) = 3$ and $\kappa(B) = 2.65 \times 10^{10}$ respectively). Thin bars hanging from the top of each plot indicate eigenvalue accuracy, displayed on the left-side y-axis.