MOX-Report No. 53/2026

Chebyshev-Filtered Randomized Low-rank Preconditioners for Symmetric Positive Definite Linear Systems

Dong Z., Jiang Y., Ng M., Ciaramella G., Yin J.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

Chebyshev-Filtered Randomized Low-rank Preconditioners for Symmetric Positive Definite Linear Systems

Zeyu Dong[1], Yurui Jiang[2], Michael Ng[3], Gabriele Ciaramella[4], Junfeng Yin[5*]

[1]Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University, Shanghai, 200092, China.
[2]School of Mathematical Sciences, Tongji University, Shanghai, 200092, China.
[3]Department of Mathematics and Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.
[4]MOX, Dipartimento di Matematica, Politecnico di Milano, Milano, 20133, Italy.
[5]Key Laboratory of Intelligent Computing and Applications (Ministry of Education), School of Mathematical Sciences, Tongji University, Shanghai, 200092, China.

*Corresponding author(s). E-mail(s): yinjf@tongji.edu.cn;
Contributing authors: dongzeyu@tongji.edu.cn; mlxg@tongji.edu.cn; michael-ng@hkbu.edu.hk; gabriele.ciaramella@polimi.it;

Abstract

Preconditioning is essential for accelerating Krylov methods for large symmetric positive definite (SPD) linear systems, especially when a small number of extremal eigenvalues deteriorate the convergence of preconditioned conjugate gradient (PCG) method. In this work, we propose a Chebyshev-filtered randomized low-rank preconditioning framework for SPD systems that targets spectral outliers at both ends of the spectrum of the coefficient matrices. The main idea is to use Chebyshev polynomial filtering to make the near-null eigenspace accessible to randomized subspace extraction. The filter amplifies the lower-tail eigencomponents that often govern PCG convergence, while randomized sketching recovers the amplified subspace using only a small number of matrix-matrix products. The resulting low-rank correction therefore targets small-eigenvalue components that

are usually missed by standard randomized subspace extraction methods. The resulting preconditioner is algebraic and admits an efficient low-rank representation. We provide subspace error bounds for the filtered randomized extraction and derive condition-number estimates for the proposed preconditioning techniques. Numerical experiments demonstrate that the proposed method improves PCG convergence, especially when small eigenvalues are the main obstruction.

Keywords: Chebyshev polynomial filtering; randomized low-rank preconditioner; extremal eigenvalues; preconditioned conjugate gradient

1 Introduction

Large sparse linear systems of the form

$$(A + \mu I)x = b \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$ symmetric positive semidefinite, $\mu \geq 0$, and $x, b \in \mathbb{R}^n$, where $A_\mu := A + \mu I$ is symmetric positive definite (SPD), arise throughout scientific computing, including finite element discretizations [1, 2], optimization subproblems [3], inverse problems [4], uncertainty quantification [5], and data-driven applications [6, 7]. The conjugate gradient (CG) method, introduced by Hestenes and Stiefel [8], is the canonical Krylov solver for (1). Its classical convergence estimate states that, after j iterations,

$$\|x^* - x_j\|_{A_\mu} \leq 2\|x^* - x_0\|_{A_\mu} \left(\frac{\sqrt{\kappa(A_\mu)} - 1}{\sqrt{\kappa(A_\mu)} + 1} \right)^j, \tag{2}$$

where x_* is the exact solution and $\kappa(A_\mu) = \lambda_{\max}(A_\mu)/\lambda_{\min}(A_\mu)$ is the spectral condition number [8, 9]. Although the practical behavior of CG depends on the full eigenvalue distribution, not only on $\kappa(A_\mu)$, (2) already highlights a central difficulty: eigenvalues close to the origin, especially a small cluster of near-null modes, can substantially slow convergence [10–12]. Therefore, improving the lower end of the spectrum remains one of the main objectives of preconditioning.

A large body of work has been devoted to one-level algebraic preconditioners, including diagonal scaling [13], incomplete factorizations [14–16], and sparse approximate inverses [17, 18]. These methods are often very effective at compressing the bulk of the spectrum and at controlling large eigenvalues, but they may still leave a small number of eigenvalues close to zero, which then dominate the Krylov convergence rate [10–12, 19, 20]. This observation has motivated two-level, coarse-space, and deflation-based strategies, whose purpose is to treat the difficult spectral components separately from the benign bulk [1, 11, 21–28]. In the ideal case, one deflates the invariant subspace associated with the smallest eigenvalues, so that the convergence is governed by an effective condition number formed from the remaining spectrum [1, 11, 12, 24].

For PDE-based problems, successful coarse spaces have been developed in multigrid, domain decomposition, and Schwarz-type frameworks [2, 29–35]. These methods are extremely powerful when geometric or analytic structure is available. However,

from the point of view of a general SPD matrix, the construction of an effective coarse space is often problem dependent. Likewise, low-rank Schur-complement corrections can be effective, but many such methods rely on graph partitioning, nested dissection, or explicit block structure, and therefore are not purely black-box algebraic procedures [12, 36–38]. Thus, despite major advances, there remains a need for preconditioners that can target the harmful low end of the spectrum while requiring only matrix-vector products or other lightweight algebraic operations.

Beyond these developments, randomized numerical linear algebra (RNLA) has emerged as a major tool for large-scale matrix computation [7, 39–41]. Randomized sketches can often approximate dominant spectral subspaces with high probability using only a small number of matrix-block products [40–43]. For symmetric positive semidefinite matrices, the Nyström approximation provides a natural randomized low-rank model [6, 41, 44], and recent work has shown that it can be used to construct practical and theoretically justified preconditioners [45]. Such methods are especially effective when the relevant spectral information is concentrated in a dominant subspace, for example at the upper end of the spectrum or when the effective dimension is modest. When CG convergence is limited by a near-null eigenspace, however, the target subspace is not dominant and is therefore much harder to capture by standard randomized low-rank approximation [40, 41].

A complementary line of work exploits polynomial filtering. Because Chebyshev polynomials are bounded on a prescribed interval and grow rapidly outside it, they provide an inexpensive mechanism for amplifying targeted spectral components through short recurrences [13, 46–49]. This idea has been used in deflation-type and spectral correction methods, where Chebyshev filters are employed to approximate invariant subspaces associated with small eigenvalues and to build partial spectral factorizations [50–54].

These observations motivate the framework developed in this paper. We propose a Chebyshev-filtered randomized low-rank preconditioner for SPD systems that targets extremal spectral components at both ends of the spectrum, with particular emphasis on near-null modes. The main idea is to use Chebyshev filtering to amplify the lower-tail eigencomponents relative to the spectral bulk, so that randomized subspace extraction can recover the corresponding invariant subspace from a small randomized sketch. This makes it possible to construct a low-rank correction for near-null modes that are typically missed by standard randomized subspace extraction methods.

To make this idea effective in practice, we further adopt a two-stage strategy. The first stage uses a cheap base preconditioner to reshape the spectrum so that most eigenvalues are compressed into a moderately clustered bulk. This spectral shaping step improves the robustness of the subsequent Chebyshev filtering, since the upper and lower tails are not numerically symmetric. The lower tail lies near the origin, whereas the upper tail may be separated from the bulk on a much larger scale; this asymmetry makes interval selection for Chebyshev filtering considerably more delicate. If the resulting polynomial amplification is too aggressive, finite-precision effects can significantly reduce the numerical robustness of the filtering process. The second stage then applies the low-rank correction to the two spectral tails in a nonuniform way. For the lower tail, we use a Chebyshev-filtered randomized sketch together with a

Rayleigh–Ritz compression to approximate the invariant subspace associated with the troublesome small eigenvalues. For the upper tail, after the spectral compression produced by the first stage, a standard randomized extraction of the dominant eigenspace is typically more stable and economical.

We derive subspace error bounds for the Chebyshev-filtered randomized extraction, showing how the recovered extremal eigenspaces depend on the separation induced by the polynomial filter and on the randomized sketch. We further show that accurate recovery of these eigenspaces leads to a reduced condition number for the practical preconditioned operator. Numerical experiments demonstrate that the proposed preconditioner is effective in practice, especially in regimes where small eigenvalues are the dominant obstacle to rapid convergence.

The remainder of the paper is organized as follows. In Section 2, we introduce notation and recall the basic properties of the Chebyshev filter. Section 3 presents the Chebyshev-filtered subspace extraction procedure together with the corresponding subspace error bounds. Section 4 introduces the oracle and practical forms of the proposed preconditioner and develops the associated condition-number analysis. Section 5 discusses implementation issues, including spectrum shaping and practical interval selection. Section 6 reports numerical experiments. Finally, Section 7 concludes the paper.

2 Preliminaries and notation

In this section, we introduce the notation used throughout the paper and recall the basic properties of Chebyshev polynomials needed later.

2.1 Spectral notation and subspace distances

Since A is symmetric positive semidefinite, it admits an eigendecomposition

$$A = U\Lambda U^\top, \qquad \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n), \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0, \tag{3}$$

where $U = [u_1, \dots, u_n]$ is orthogonal. For an index set $\mathcal{J} \subset \{1, \dots, n\}$, we write $U_{\mathcal{J}}$ for the matrix whose columns are $\{u_j\}_{j\in\mathcal{J}}$, and $\Lambda_{\mathcal{J}}$ for the corresponding diagonal block of eigenvalues.

When both ends of the spectrum are of interest, we use the index set

$$\mathcal{J} = \{1, \dots, \ell_1\} \cup \{n - \ell_2 + 1, \dots, n\},$$

and denote by $U_\ell = [u_1, \dots, u_{\ell_1}, u_{n-\ell_2+1}, \dots, u_n] \in \mathbb{R}^{n\times\ell}$, with $\ell = \ell_1 + \ell_2$, the exact invariant subspace associated with the ℓ_1 largest and ℓ_2 smallest eigenvalues.

For any matrix $Q \in \mathbb{R}^{n\times r}$ with orthonormal columns, let $\Pi_Q := QQ^\top$ denote the orthogonal projector onto range(Q). In particular, $\Pi := U_\ell U_\ell^\top$ denotes the projector onto the exact extremal eigenspace. If $\hat{U}_\ell$ is an approximate basis, we write $\hat{\Pi} := \hat{U}_\ell \hat{U}_\ell^\top$ for the associated projector. Throughout the paper, symbols carrying a hat denote approximate quantities, while the corresponding unhatted symbols refer to the exact spectral objects.

To measure the discrepancy between subspaces, we use principal angles. For two subspaces $\mathcal{S}$ and $\mathcal{T}$ of the same dimension, $\Theta(\mathcal{S}, \mathcal{T})$ denotes the diagonal matrix of principal angles. When $\sin\Theta(\mathcal{S}, \mathcal{T})$ or $\tan\Theta(\mathcal{S}, \mathcal{T})$ appears as a scalar quantity, it denotes the largest sine or tangent of the principal angles. With this convention, we repeatedly use the identity

$$\|\Pi_{Q_1} - \Pi_{Q_2}\|_2 = \sin\Theta(\mathrm{range}(Q_1), \mathrm{range}(Q_2)),$$

which allows one to pass between projector perturbation and subspace error. We also use $\tan\Theta(\mathcal{S}, \mathcal{T})$ to measure the largest tangent-angle between subspaces.

We conclude this subsection with two standard facts on Gaussian test matrices that will be used later in the expectation bound for the filtered randomized subspace error; see Section 10 of Halko, Martinsson, and Tropp [40]. The first gives a bound for the expected norm of a scaled Gaussian matrix, and the second provides standard estimates for the pseudoinverse under oversampling.

Lemma 1 *Let S and T be fixed matrices of compatible dimensions, and let G be a standard Gaussian matrix. Then*

$$\mathbb{E}\|SGT\|_2 \le \|S\|_2 \|T\|_F + \|S\|_F \|T\|_2. \tag{4}$$

Lemma 2 *Let $G \in \mathbb{R}^{k\times(k+p)}$ be a standard Gaussian matrix with $k \ge 2$ and $p \ge 2$. Then*

$$\left(\mathbb{E}\|G^\dagger\|_F^2\right)^{1/2} = \sqrt{\frac{k}{p-1}}, \tag{5}$$

$$\mathbb{E}\|G^\dagger\|_2 \le \frac{e\sqrt{k+p}}{p}. \tag{6}$$

2.2 Chebyshev polynomial filter

Let T_m be the Chebyshev polynomial of the first kind of degree m, defined by

$$T_0(x) = 1, \qquad T_1(x) = x, \qquad T_{m+1}(x) = 2xT_m(x) - T_{m-1}(x), \qquad m \ge 1.$$

Given an interval $[a, b]$ with $a < b$, define the affine map

$$\phi(\lambda) = \frac{\lambda - \frac{a+b}{2}}{\frac{b-a}{2}}, \qquad p_m(\lambda) = T_m(\phi(\lambda)). \tag{7}$$

By construction, ϕ maps $[a, b]$ onto $[-1, 1]$. Hence

$$|p_m(\lambda)| \le 1, \qquad \lambda \in [a, b],$$

whereas $|p_m(\lambda)|$ grows rapidly when $|\phi(\lambda)| > 1$. This separation property is the basis of Chebyshev filtering. When $[a, b]$ contains the non-target part of the spectrum, the polynomial remains bounded on the bulk spectrum and amplifies the components associated with the target eigenvalues outside the interval.

3 Chebyshev-filtered randomized subspace extraction

Before presenting the extraction procedure, we first explain the role of the Chebyshev filter. In principle, the subspace extraction argument below applies to any polynomial filter. What matters for the approximation is the separation between the filter gain on the target spectral region and that on the non-target region. A useful filter should therefore remain small on the non-target interval while strongly amplifying the desired eigencomponents outside that interval.

Chebyshev polynomials provide a natural choice for this purpose. After an affine map sends the non-target interval to $[-1, 1]$, the Chebyshev polynomial is uniformly bounded on this interval and grows rapidly outside it. This property has the classical minimax interpretation that, for a prescribed exterior point $|\eta| > 1$, the normalized polynomial $T_m(x)/T_m(\eta)$ minimizes

$$\max_{x\in[-1,1]} |q(x)|$$

among all polynomials q of degree at most m satisfying $q(\eta) = 1$. Equivalently, under a boundedness constraint on the non-target interval, the Chebyshev filter gives extremal amplification at an exterior target point.

The same choice is also computationally attractive. Chebyshev polynomials satisfy a three-term recurrence, so the action of $p_m(A)$ on a block of vectors can be evaluated without forming either the polynomial or powers of A explicitly. The filtering step therefore requires only matrix-block products and short recurrences, making it well suited for large sparse matrices and matrix-free implementations.

3.1 Construction of the filtered randomized subspace extraction

In this section, we introduce the basic randomized filtering procedure that underlies the proposed preconditioner. The purpose of this step is to construct, from a random sketch and a Chebyshev polynomial filter, a low-dimensional subspace that approximates an invariant subspace associated with a prescribed spectral region. This construction will later be applied to the large- and small-eigenvalue tails of the spectrum.

Let $[a, b]$ be an interval containing the non-target spectrum, and let p_m be the Chebyshev polynomial filter associated with this interval. Since the filter remains bounded on $[a, b]$ and grows outside it, applying $p_m(A)$ to a random test matrix enhances the components associated with the target eigenvalues while suppressing the bulk contribution. The resulting procedure is summarized in Algorithm 1.

Algorithm 1 Chebyshev-filtered randomized subspace extraction

Require: Symmetric positive semidefinite matrix $A \in \mathbb{R}^{n\times n}$, extraction dimension ℓ, polynomial degree $m \geq 0$, interval $[a, b]$ containing the non-target spectrum.
Ensure: Ritz vectors $\hat{U}_\ell \in \mathbb{R}^{n\times \ell}$ and Ritz values $\hat{\Lambda}_\ell$.
1: Draw a random matrix $\Omega \in \mathbb{R}^{n\times \ell}$
2: Define

$$\phi(\lambda) = \frac{\lambda - \frac{a+b}{2}}{\frac{b-a}{2}}, \qquad p_m(\lambda) = T_m\big(\phi(\lambda)\big),$$

where T_m is the Chebyshev polynomial of the first kind
3: Form the filtered sketch $Y = p_m(A)\Omega$ by the three-term Chebyshev recurrence
4: Compute a thin QR factorization $Y = Q_\ell R$
5: Form $T = Q_\ell^\top A Q_\ell$ and compute its eigendecomposition $T = W_\ell \hat{\Lambda}_\ell W_\ell^\top$
6: Set $\hat{U}_\ell = Q_\ell W_\ell$

3.2 Subspace error bounds

We now quantify this approximation property. The following bound is stated for a general polynomial filter p_m. The Chebyshev choice will be used afterward to estimate the gain ratio β_1/β_2.

Theorem 3 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive semidefinite with eigendecomposition $A = U\Lambda U^\top$, where $U = [U_1, U_2]$ and $\Lambda = \mathrm{blkdiag}(\Lambda_1, \Lambda_2)$. Here $\Lambda_1 \in \mathbb{R}^{k\times k}$ contains the target eigenvalues and Λ_2 contains the remaining eigenvalues. Let $\Omega \in \mathbb{R}^{n\times \ell}$ with $\ell \geq k$, set $\Omega_1 = U_1^\top \Omega$ and $\Omega_2 = U_2^\top \Omega$, and define $Y = p_m(A)\Omega$ and $Q = \mathrm{orth}(Y)$. Assume that Ω_1 has full row rank, and define*

$$\beta_1 = \sigma_{\max}\big(p_m(\Lambda_2)\big), \qquad \beta_2 = \sigma_{\min}\big(p_m(\Lambda_1)\big).$$

If $\beta_2 > 0$, then the k principal angles between $\mathrm{range}(U_1)$ and $\mathrm{range}(Q)$ satisfy

$$\tan\Theta\big(\mathrm{range}(U_1), \mathrm{range}(Q)\big) \leq \frac{\beta_1}{\beta_2}\|\Omega_2\Omega_1^\dagger\|_2.$$

Consequently, the same bound holds with $\mathrm{range}(Q)$ replaced by $\mathrm{range}(\hat{U}_\ell)$.

Proof Since p_m is a polynomial, one has

$$p_m(A) = U \begin{bmatrix} p_m(\Lambda_1) & 0 \\ 0 & p_m(\Lambda_2) \end{bmatrix} U^\top.$$

Therefore, it holds that

$$Y = p_m(A)\Omega = U_1 p_m(\Lambda_1)\Omega_1 + U_2 p_m(\Lambda_2)\Omega_2,$$

and introducing the shorthand notation

$$G := p_m(\Lambda_1)\Omega_1, \qquad H := p_m(\Lambda_2)\Omega_2,$$

one gets

$$Y = U_1 G + U_2 H.$$

Because $\beta_2 > 0$, the matrix $p_m(\Lambda_1)$ is invertible. Since Ω_1 has full row rank by assumption, it follows that G also has full row rank k. Hence its Moore–Penrose pseudoinverse $G^\dagger$ is well defined and satisfies

$$GG^\dagger = I_k.$$

Now, define

$$F := HG^\dagger \in \mathbb{R}^{(n-k)\times k}, \qquad Z := YG^\dagger.$$

Using $Y = U_1G + U_2H$ and $GG^\dagger = I_k$, we obtain

$$Z = YG^\dagger = U_1GG^\dagger + U_2HG^\dagger = U_1 + U_2F.$$

One has that,

$$\operatorname{range}(Z) \subseteq \operatorname{range}(Y) = \operatorname{range}(Q).$$

We next show that $\operatorname{range}(Z)$ is a k-dimensional graph subspace over $\operatorname{range}(U_1)$. Indeed,

$$U_1^\top Z = U_1^\top (U_1 + U_2F) = I_k,$$

and thus Z has full column rank k. Let

$$\hat{Q} := \operatorname{orth}(Z) = Z(Z^\top Z)^{-1/2},$$

so that $\operatorname{range}(\hat{Q}) = \operatorname{range}(Z)$. Since $U_1^\top U_1 = I$, $U_2^\top U_2 = I$, and $U_1^\top U_2 = 0$, we have

$$Z^\top Z = (U_1 + U_2F)^\top (U_1 + U_2F) = I_k + F^\top F.$$

Consequently,

$$U_1^\top \hat{Q} = U_1^\top Z(Z^\top Z)^{-1/2} = (I_k + F^\top F)^{-1/2}, \tag{8}$$

and

$$U_2^\top \hat{Q} = U_2^\top Z(Z^\top Z)^{-1/2} = F(I_k + F^\top F)^{-1/2}. \tag{9}$$

Since $U_1^\top \hat{Q}$ is invertible, the tangent-angle representation for two k-dimensional subspaces gives

$$\tan\Theta(\operatorname{range}(U_1), \operatorname{range}(\hat{Q})) = \left\| U_2^\top \hat{Q}\, (U_1^\top \hat{Q})^{-1} \right\|_2.$$

Using (8) and (9) yields

$$U_2^\top \hat{Q}\, (U_1^\top \hat{Q})^{-1} = F(I_k + F^\top F)^{-1/2}(I_k + F^\top F)^{1/2} = F,$$

and hence

$$\tan\Theta(\operatorname{range}(U_1), \operatorname{range}(\hat{Q})) = \|F\|_2.$$

Now $\operatorname{range}(\hat{Q}) = \operatorname{range}(Z)$ is a k-dimensional subspace contained in $\operatorname{range}(Q)$. Therefore, the k principal angles between $\operatorname{range}(U_1)$ and $\operatorname{range}(Q)$ cannot exceed the corresponding principal angles between $\operatorname{range}(U_1)$ and $\operatorname{range}(\hat{Q})$. Thus

$$\tan\Theta(\operatorname{range}(U_1), \operatorname{range}(Q)) \le \tan\Theta(\operatorname{range}(U_1), \operatorname{range}(\hat{Q})) = \|F\|_2.$$

It remains to bound $\|F\|_2$. Since $p_m(\Lambda_1)$ is invertible and Ω_1 has full row rank, the standard pseudoinverse identity gives

$$G^\dagger = \big(p_m(\Lambda_1)\Omega_1\big)^\dagger = \Omega_1^\dagger\, p_m(\Lambda_1)^{-1}.$$

By construction $F = HG^\dagger$, it holds that

$$\|F\|_2 = \|p_m(\Lambda_2)\Omega_2\Omega_1^\dagger\, p_m(\Lambda_1)^{-1}\|_2 \le \frac{\beta_1}{\beta_2}\|\Omega_2\Omega_1^\dagger\|_2.$$

Together with the previous inequality, this proves

$$\tan\Theta\big(\operatorname{range}(U_1), \operatorname{range}(Q)\big) \le \frac{\beta_1}{\beta_2}\,\|\Omega_2\Omega_1^\dagger\|_2.$$

The proof is complete. □

Theorem 3 shows that the approximation error is controlled by the ratio between the filter gain on the non-target block and the minimum gain on the target block. The Chebyshev filter is well suited for reducing this ratio because, in the classical minimax sense, it provides optimal polynomial separation between the non-target interval and exterior target eigenvalues.

Theorem 4 *Assume the hypotheses of Theorem 3. For the Chebyshev filter* $p_m(\lambda) = T_m(\phi(\lambda))$ *used in Algorithm 1, assume that the non-target eigenvalues lie in* $[a, b]$ *and (recall (7)) that* $|\phi(\lambda)| \geq \eta > 1$ *for all* $\lambda \in \lambda(\Lambda_1)$*, then*

$$\beta_2 \geq \frac{1}{2}\big(\eta + \sqrt{\eta^2 - 1}\big)^m,$$

and hence

$$\tan\Theta\big(\mathrm{range}(U_1), \mathrm{range}(Q)\big) \leq 2\big(\eta + \sqrt{\eta^2 - 1}\big)^{-m} \|\Omega_2 \Omega_1^\dagger\|_2.$$

Proof For $|x| \geq \eta > 1$,

$$|T_m(x)| = \cosh\big(m \operatorname{arcosh}(|x|)\big) \geq \frac{1}{2}\exp\big(m \operatorname{arcosh}(\eta)\big) = \frac{1}{2}\big(\eta + \sqrt{\eta^2 - 1}\big)^m.$$

Taking the minimum over $\lambda \in \lambda(\Lambda_1)$ gives the lower bound for β_2. Since the non-target eigenvalues lie in $[a, b]$, we also have $\beta_1 = \sigma_{\max}\big(p_m(\Lambda_2)\big) \leq 1$. The result follows from Theorem 3. □

Theorem 4 shows that, for a fixed realization of the sketch, the subspace error decays exponentially with the polynomial degree once the target spectrum lies outside the Chebyshev interval. For the randomized algorithm, however, the remaining quantity to control is the random factor $\|\Omega_2 \Omega_1^\dagger\|_2$. We therefore turn to its expectation under a Gaussian test matrix.

Lemma 5 *Let* $\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix}, \Omega_1 \in \mathbb{R}^{k \times (k+p)}, \Omega_2 \in \mathbb{R}^{(n-k) \times (k+p)}$*, where* Ω *is a standard Gaussian matrix and* $p \geq 2$*. Then*

$$\mathbb{E}\|\Omega_2 \Omega_1^\dagger\|_2 \leq \sqrt{\frac{k}{p-1}} + \sqrt{n-k}\,\frac{e\sqrt{k+p}}{p}.$$

Proof By rotational invariance of the Gaussian distribution, Ω_1 and Ω_2 are independent standard Gaussian matrices of sizes $k \times (k+p)$ and $(n-k) \times (k+p)$, respectively. Conditioning on Ω_1 and applying (4) with $S = I_{n-k}$, $G = \Omega_2$ and $T = \Omega_1^\dagger$, we obtain

$$\mathbb{E}_{\Omega_2}\Big[\|\Omega_2 \Omega_1^\dagger\|_2 \,\Big|\, \Omega_1\Big] \leq \|I_{n-k}\|_2 \,\|\Omega_1^\dagger\|_F + \|I_{n-k}\|_F \,\|\Omega_1^\dagger\|_2 = \|\Omega_1^\dagger\|_F + \sqrt{n-k}\,\|\Omega_1^\dagger\|_2.$$

Taking expectation with respect to Ω_1 yields

$$\mathbb{E}\|\Omega_2 \Omega_1^\dagger\|_2 \leq \mathbb{E}\|\Omega_1^\dagger\|_F + \sqrt{n-k}\,\mathbb{E}\|\Omega_1^\dagger\|_2.$$

Since $\Omega_1 \in \mathbb{R}^{k \times (k+p)}$ is standard Gaussian with $p \geq 2$, Lemma 2 gives

$$\Big(\mathbb{E}\|\Omega_1^\dagger\|_F^2\Big)^{1/2} = \sqrt{\frac{k}{p-1}}, \qquad \mathbb{E}\|\Omega_1^\dagger\|_2 \leq \frac{e\sqrt{k+p}}{p}.$$

By Jensen's inequality,

$$\mathbb{E}\|\Omega_1^\dagger\|_F \le \left(\mathbb{E}\|\Omega_1^\dagger\|_F^2\right)^{1/2} = \sqrt{\frac{k}{p-1}}.$$

Combining the above bounds proves the result. □

Theorem 6 *Assume the hypotheses of Theorem 3. Let $\ell = k+p$ with $p \ge 2$, and let $\Omega \in \mathbb{R}^{n\times\ell}$ be a standard Gaussian matrix. Then*

$$\mathbb{E}\big[\tan\Theta(\mathrm{range}(U_1), \mathrm{range}(Q))\big] \le \frac{\beta_1}{\beta_2}\left(\sqrt{\frac{k}{p-1}} + \sqrt{n-k}\,\frac{e\sqrt{\ell}}{p}\right).$$

If, in addition, $p_m(\lambda) = T_m(\phi(\lambda))$ is the Chebyshev filter used in Algorithm 1, $|\phi(\lambda)| \ge \eta > 1$ for all $\lambda \in \lambda(\Lambda_1)$, and $\beta_1 \le 1$, then

$$\mathbb{E}\big[\tan\Theta(\mathrm{range}(U_1), \mathrm{range}(Q))\big] \le 2\left(\eta + \sqrt{\eta^2-1}\right)^{-m}\left(\sqrt{\frac{k}{p-1}} + \sqrt{n-k}\,\frac{e\sqrt{\ell}}{p}\right).$$

Proof By Theorem 3,

$$\tan\Theta(\mathrm{range}(U_1), \mathrm{range}(Q)) \le \frac{\beta_1}{\beta_2}\,\|\Omega_2\Omega_1^\dagger\|_2.$$

Taking expectation and applying Lemma 5 yields

$$\mathbb{E}\big[\tan\Theta(\mathrm{range}(U_1), \mathrm{range}(Q))\big] \le \frac{\beta_1}{\beta_2}\,\mathbb{E}\|\Omega_2\Omega_1^\dagger\|_2 \le \frac{\beta_1}{\beta_2}\left(\sqrt{\frac{k}{p-1}} + \sqrt{n-k}\,\frac{e\sqrt{\ell}}{p}\right).$$

For the Chebyshev filter, Theorem 4 gives

$$\frac{\beta_1}{\beta_2} \le 2\left(\eta + \sqrt{\eta^2-1}\right)^{-m}.$$

Substituting this estimate into the first bound proves the second claim. □

Theorem 6 controls the subspace error in expectation. Since the practical preconditioner is constructed from a single realized sketch, we also record the following high-probability counterpart.

Theorem 7 *Assume the hypotheses of Theorem 3, and let $\ell = k+p$. Let $\Omega \in \mathbb{R}^{n\times\ell}$ be a standard Gaussian matrix. For any $t_1 > 0$ and any $0 < t_2 < \sqrt{\ell} - \sqrt{k}$, with probability at least $1 - \exp(-t_1^2/2) - \exp(-t_2^2/2)$, one has*

$$\tan\Theta\big(\mathrm{range}(U_1), \mathrm{range}(Q)\big) \le \frac{\beta_1}{\beta_2}\,\frac{\sqrt{n-k} + \sqrt{\ell} + t_1}{\sqrt{\ell} - \sqrt{k} - t_2}.$$

If, in addition, $p_m(\lambda) = T_m(\phi(\lambda))$ is the Chebyshev filter used in Algorithm 1, $|\phi(\lambda)| \ge \eta > 1$ for all $\lambda \in \lambda(\Lambda_1)$ and $\beta_1 \le 1$, then, with the same probability,

$$\tan\Theta\big(\mathrm{range}(U_1), \mathrm{range}(Q)\big) \le 2\left(\eta + \sqrt{\eta^2-1}\right)^{-m}\,\frac{\sqrt{n-k} + \sqrt{\ell} + t_1}{\sqrt{\ell} - \sqrt{k} - t_2}.$$

Proof By Theorem 3,

$$\tan\Theta\big(\mathrm{range}(U_1),\mathrm{range}(Q)\big) \le \frac{\beta_1}{\beta_2}\|\Omega_2\Omega_1^\dagger\|_2.$$

It remains to derive a high-probability bound for $\|\Omega_2\Omega_1^\dagger\|_2$. By rotational invariance of the Gaussian distribution, Ω_1 and Ω_2 are independent standard Gaussian matrices of sizes $k\times\ell$ and $(n-k)\times\ell$, respectively. We first condition on Ω_1. For fixed Ω_1, define

$$f(G) = \|G\Omega_1^\dagger\|_2.$$

Then f is $\|\Omega_1^\dagger\|_2$-Lipschitz with respect to the Frobenius norm, since

$$|f(G)-f(H)| \le \|(G-H)\Omega_1^\dagger\|_2 \le \|G-H\|_F\|\Omega_1^\dagger\|_2.$$

By the Gaussian concentration inequality for Lipschitz functions [55, Theorem 5.6], if f is L-Lipschitz, then

$$\mathbb{P}\{f(G)\le \mathbb{E}f(G)+Lt\} \ge 1-\exp(-t^2/2).$$

Applying this bound with $G=\Omega_2$, $L=\|\Omega_1^\dagger\|_2$, and $t=t_1$, for every $t_1>0$, conditioning on Ω_1, we obtain

$$\mathbb{P}_{\Omega_2}\left\{\|\Omega_2\Omega_1^\dagger\|_2 \le \mathbb{E}_{\Omega_2}\left[\|\Omega_2\Omega_1^\dagger\|_2\,\middle|\,\Omega_1\right] + t_1\|\Omega_1^\dagger\|_2\,\middle|\,\Omega_1\right\} \ge 1-\exp(-t_1^2/2). \tag{10}$$

Applying (4) with $S=I_{n-k}$, $G=\Omega_2$ and $T=\Omega_1^\dagger$, we obtain

$$\mathbb{E}_{\Omega_2}\left[\|\Omega_2\Omega_1^\dagger\|_2\,\middle|\,\Omega_1\right] \le \|\Omega_1^\dagger\|_F + \sqrt{n-k}\,\|\Omega_1^\dagger\|_2. \tag{11}$$

Therefore, combining (10) and (11), we obtain, with conditional probability at least $1-\exp(-t_1^2/2)$,

$$\|\Omega_2\Omega_1^\dagger\|_2 \le \|\Omega_1^\dagger\|_F + (\sqrt{n-k}+t_1)\|\Omega_1^\dagger\|_2.$$

Since $\Omega_1^\dagger\in\mathbb{R}^{\ell\times k}$, we have

$$\|\Omega_1^\dagger\|_F \le \sqrt{\ell}\,\|\Omega_1^\dagger\|_2.$$

Thus,

$$\|\Omega_2\Omega_1^\dagger\|_2 \le (\sqrt{n-k}+\sqrt{\ell}+t_1)\|\Omega_1^\dagger\|_2.$$

It remains to control $\|\Omega_1^\dagger\|_2$. The standard lower-tail bound for the smallest singular value of a rectangular Gaussian matrix [56] gives

$$\mathbb{P}\left\{\sigma_{\min}(\Omega_1) \ge \sqrt{\ell}-\sqrt{k}-t_2\right\} \ge 1-\exp(-t_2^2/2).$$

On this event,

$$\|\Omega_1^\dagger\|_2 = \frac{1}{\sigma_{\min}(\Omega_1)} \le \frac{1}{\sqrt{\ell}-\sqrt{k}-t_2}.$$

Combining the two estimates and applying the union bound yields

$$\|\Omega_2\Omega_1^\dagger\|_2 \le \frac{\sqrt{n-k}+\sqrt{\ell}+t_1}{\sqrt{\ell}-\sqrt{k}-t_2}$$

with probability at least $1-\exp(-t_1^2/2)-\exp(-t_2^2/2)$. Substituting this bound into the deterministic estimate proves

$$\tan\Theta\big(\mathrm{range}(U_1),\mathrm{range}(Q)\big) \le \frac{\beta_1}{\beta_2}\frac{\sqrt{n-k}+\sqrt{\ell}+t_1}{\sqrt{\ell}-\sqrt{k}-t_2}.$$

For the Chebyshev filter, Theorem 4 gives

$$\frac{\beta_1}{\beta_2} \le 2\left(\eta+\sqrt{\eta^2-1}\right)^{-m}.$$

Substituting this estimate into the first bound proves the second claim. □

We next relate the subspace-angle bounds above to the associated projector error.

Remark 8 For two subspaces of equal dimension,

$$\|U_\ell U_\ell^\top - \hat{U}_\ell \hat{U}_\ell^\top\|_2 = \sin\Theta(\mathrm{range}(U_\ell), \mathrm{range}(\hat{U}_\ell)).$$

Since $\sin\theta = \tan\theta/\sqrt{1+\tan^2\theta}$ for every principal angle θ, one has $\sin\Theta \le \tan\Theta$. Therefore, any bound on $\tan\Theta(\mathrm{range}(\hat{U}_\ell), \mathrm{range}(U_\ell))$ immediately yields a bound of the same order for the projector error $\|U_\ell U_\ell^\top - \hat{U}_\ell \hat{U}_\ell^\top\|_2$.

4 Low-rank preconditioning for extremal spectral components

We now turn from subspace extraction to the construction of the preconditioner itself. The analysis in Section 3 shows that a Chebyshev-filtered randomized sketch can recover, to good accuracy, the invariant subspaces associated with the extreme eigenvalues. Once such subspaces have been identified, a natural strategy is to treat them explicitly and approximate the remaining spectral components by a scalar term.

This point of view leads to a subspace-complement construction. On the extracted extremal subspace, the preconditioner matches A_μ on the corresponding invariant directions, since these modes are precisely the ones that are most harmful for iterative convergence. On the orthogonal complement, where the spectrum is expected to be more benign, a scalar approximation is used to keep the representation inexpensive. We first analyze this idea in an oracle setting with the exact extremal invariant subspace, and then replace it by the subspace obtained from the filtered randomized procedure.

4.1 Oracle preconditioners

To make the subspace-complement structure precise, we begin with the ideal situation in which the exact extremal invariant subspace is available. This oracle construction serves as a benchmark for the practical preconditioner introduced later.

Definition 9 Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Let $A = U\Lambda U^\top$ be an eigendecomposition, where $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ and $\lambda_1 \ge \cdots \ge \lambda_n$. Let $\mathcal{J} = \{1, \dots, \ell_1\} \cup \{n-\ell_2+1, \dots, n\}$ be the set of captured extreme indices, and let $U_\ell \in \mathbb{R}^{n\times\ell}$, with $\ell = \ell_1 + \ell_2$, collect the corresponding eigenvectors. Let $\Lambda_\ell = \mathrm{diag}(\{\lambda_i\}_{i\in\mathcal{J}})$. For $\alpha > 0$, define the oracle preconditioner for A_μ as

$$P_*^{(A_\mu)} = U_\ell(\Lambda_\ell + \mu I_\ell)U_\ell^\top + \alpha(I - U_\ell U_\ell^\top). \tag{12}$$

By construction, $P_*^{(A_\mu)}$ agrees with A_μ on $\mathrm{range}(U_\ell)$ and replaces the action of A_μ on $\mathrm{range}(U_\ell)^\perp$ by the scalar α. This gives the following explicit description of the preconditioned spectrum.

Proposition 10 *Under the setting of Definition 9,*

$$\mathrm{spec}\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2}\Big) = \underbrace{\{1, \dots, 1\}}_{\ell \text{ times}} \cup \left\{\frac{\lambda_i + \mu}{\alpha} : i \notin \mathcal{J}\right\}.$$

Proof For each $i \in \mathcal{J}$, the vector u_i belongs to range(U_ℓ), so

$$P_*^{(A_\mu)} u_i = (\lambda_i + \mu) u_i.$$

Hence

$$\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2} u_i = u_i.$$

For each $i \notin \mathcal{J}$, the vector u_i lies in range$(U_\ell)^\perp$, and therefore

$$P_*^{(A_\mu)} u_i = \alpha u_i.$$

It follows that

$$\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2} u_i = \frac{\lambda_i + \mu}{\alpha} u_i.$$

Since the eigenvectors of A form an orthonormal basis, the claim follows. □

Proposition 10 shows the spectral effect of the oracle preconditioner. The captured extremal eigenvalues are mapped exactly to one, whereas the remaining eigenvalues are scaled by the single scalar α. Consequently, the condition number is determined by the spread of $(\lambda_i + \mu)/\alpha$ over the uncaptured indices. This observation immediately yields a condition number formula and an interval-based bound.

Corollary 11 *Assume $A_\mu \succ 0$ and $\alpha > 0$. Then*

$$\kappa_2\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2}\Big) = \frac{\max\left\{1, \max_{i \notin \mathcal{J}} \frac{\lambda_i + \mu}{\alpha}\right\}}{\min\left\{1, \min_{i \notin \mathcal{J}} \frac{\lambda_i + \mu}{\alpha}\right\}}. \tag{13}$$

Moreover, if $\lambda_i + \mu \in [a_\mu, b_\mu]$ for all $i \notin \mathcal{J}$, then

$$\kappa_2\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2}\Big) \le \frac{\max\{1, b_\mu/\alpha\}}{\min\{1, a_\mu/\alpha\}}. \tag{14}$$

The remaining question is how to choose the scalar α. In principle, the best choice depends on the uncaptured middle spectrum. In practice, however, the inner endpoints of this spectral block are typically unknown, whereas the outer endpoints adjacent to the extracted extremal blocks are available from Ritz information. It is therefore preferable to base the choice of α on quantities that can actually be computed. Let $\mathcal{I}_{\mathrm{mid}} = \{\ell_1 + 1, \dots, n - \ell_2\}$. Then for every $i \in \mathcal{I}_{\mathrm{mid}}$, $\lambda_{\ell_1} \ge \lambda_i \ge \lambda_{n-\ell_2+1}$, so the middle spectrum is contained in the computable interval $[\lambda_{n-\ell_2+1}, \lambda_{\ell_1}]$. Setting $a_\mu = \lambda_{n-\ell_2+1} + \mu$ and $b_\mu = \lambda_{\ell_1} + \mu$, with $0 < a_\mu \le b_\mu$, we obtain

$$\max_{i \in \mathcal{I}_{\mathrm{mid}}} \left| \frac{1}{\lambda_i + \mu} - \frac{1}{\alpha} \right| \le \max_{x \in [a_\mu, b_\mu]} \left| \frac{1}{x} - \frac{1}{\alpha} \right|. \tag{15}$$

Thus the choice of α can be reduced to a one-dimensional approximation problem on the interval $[a_\mu, b_\mu]$.

Two choices are particularly natural. The first is the harmonic-mean choice

$$\alpha_{\mathrm{harm}} = \frac{2 a_\mu b_\mu}{a_\mu + b_\mu}, \tag{16}$$

which minimizes the maximum absolute deviation of $1/\alpha$ from $1/x$ over $[a_\mu, b_\mu]$. The second is the geometric-mean choice

$$\alpha_{\text{geo}} = \sqrt{a_\mu b_\mu}, \tag{17}$$

which balances the ratio x/α over the same interval. Both choices lie in $[a_\mu, b_\mu]$, and therefore both lead to the same condition-number upper bound

$$\kappa_2\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2}\Big) \le \frac{b_\mu}{a_\mu}. \tag{18}$$

4.2 Practical preconditioner

In the practical method, the exact basis U_ℓ is unavailable and is replaced by the Ritz basis produced by the Chebyshev-filtered randomized extraction. In the notation of Algorithm 1, after the upper and lower spectral components have been extracted and combined, we write the resulting extremal Ritz pair as $\hat{U}_\ell \in \mathbb{R}^{n\times\ell}$, $\hat{U}_\ell^\top \hat{U}_\ell = I_\ell$, and $\hat{U}_\ell^\top A \hat{U}_\ell = \hat{\Lambda}_\ell$, where $\ell = \ell_1 + \ell_2$ and $\hat{\Lambda}_\ell \in \mathbb{R}^{\ell\times\ell}$ is diagonal. We then define the computable Chebyshev-based preconditioner by

$$P_{\text{che}}^{(A_\mu)} = \hat{U}_\ell(\hat{\Lambda}_\ell + \mu I_\ell)\hat{U}_\ell^\top + \alpha(I - \hat{U}_\ell\hat{U}_\ell^\top). \tag{19}$$

The interpretation of (19) is the same as in the oracle case. On the approximate extremal subspace range($\hat{U}_\ell$), the preconditioner uses the Ritz approximation of the regularized operator, while on the orthogonal complement it is represented by the scalar α. A useful consequence of this subspace-complement form (19) is that the inverse of the preconditioner admits a closed-form expression

$$\big(P_{\text{che}}^{(A_\mu)}\big)^{-1} = \hat{U}_\ell(\hat{\Lambda}_\ell + \mu I_\ell)^{-1}\hat{U}_\ell^\top + \alpha^{-1}(I - \hat{U}_\ell\hat{U}_\ell^\top). \tag{20}$$

4.3 Approximation and condition-number analysis

We now compare the practical preconditioner with the oracle model and quantify how well it approximates the action of A_μ^{-1}. The following result quantifies the error in approximating A_μ^{-1} by $\big(P_{\text{che}}^{(A_\mu)}\big)^{-1}$. The bound contains two contributions: a deterministic mid-spectrum term, which reflects the scalar approximation on the uncaptured part of the spectrum, and a projector perturbation term, which accounts for replacing the exact extremal eigenspace by its Ritz approximation.

Before stating the next result, we define the exact and approximate projectors $\Pi = U_\ell U_\ell^\top$ and $\hat{\Pi} = \hat{U}_\ell\hat{U}_\ell^\top$, respectively. We also introduce the auxiliary matrices

$$\hat{A}_{\text{che}} = \hat{U}_\ell\hat{\Lambda}_\ell\hat{U}_\ell^\top + (\alpha-\mu)(I-\hat{\Pi}), \qquad A_\ell = U_\ell\Lambda_\ell U_\ell^\top + (\alpha-\mu)(I-\Pi).$$

These matrices are used in the proof of Theorem 12 to compare the practical preconditioner with the corresponding oracle model.

Theorem 12 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive semidefinite, and let $\mu > 0$. Let $A = U\Lambda U^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ and $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. Fix integers ℓ_1 and ℓ_2, set $\ell = \ell_1 + \ell_2$, define $U_\ell = [u_1, \dots, u_{\ell_1}, u_{n-\ell_2+1}, \dots, u_n] \in \mathbb{R}^{n\times \ell}$. Let $\Lambda_\ell = \mathrm{diag}(\lambda_1, \dots, \lambda_{\ell_1}, \lambda_{n-\ell_2+1}, \dots, \lambda_n)$. Let $(\hat{U}_\ell, \hat{\Lambda}_\ell)$ be the extremal Ritz pair obtained from the filtered randomized extraction, where $\hat{U}_\ell \in \mathbb{R}^{n\times\ell}$ has orthonormal columns and $\hat{\Lambda}_\ell = \hat{U}_\ell^\top A \hat{U}_\ell$. Assume that $\alpha \geq \mu$ and*

$$\tan\Theta(\mathrm{range}(U_\ell), \mathrm{range}(\hat{U}_\ell)) \leq \delta.$$

Let $\mathcal{I}_{\mathrm{mid}} = \{\ell_1 + 1, \dots, n - \ell_2\}$. Then

$$\|(A+\mu I)^{-1} - (P_{\mathrm{che}}^{(A_\mu)})^{-1}\|_2 \leq \max_{i\in\mathcal{I}_{\mathrm{mid}}} \left|\frac{1}{\lambda_i+\mu} - \frac{1}{\alpha}\right| + \frac{2\|A\|_2 + |\alpha-\mu|}{\mu^2}\,\delta. \tag{21}$$

Proof By the triangle inequality, it holds that

$$\|(A+\mu I)^{-1} - (P_{\mathrm{che}}^{(A_\mu)})^{-1}\|_2 \leq \|(A+\mu I)^{-1} - (A_\ell+\mu I)^{-1}\|_2 + \|(A_\ell+\mu I)^{-1} - (P_{\mathrm{che}}^{(A_\mu)})^{-1}\|_2.$$

We estimate the two terms on the right-hand side separately. We first estimate the second term. By the definition (19), one has $P_{\mathrm{che}}^{(A_\mu)} = \hat{A}_{\mathrm{che}} + \mu I$. Since $\hat{\Lambda}_\ell = \hat{U}_\ell^\top A \hat{U}_\ell$, it holds that

$$\hat{U}_\ell \hat{\Lambda}_\ell \hat{U}_\ell^\top = \hat{U}_\ell(\hat{U}_\ell^\top A \hat{U}_\ell)\hat{U}_\ell^\top = \hat{\Pi} A \hat{\Pi}.$$

Therefore

$$\hat{A}_{\mathrm{che}} = \hat{\Pi} A \hat{\Pi} + (\alpha-\mu)(I - \hat{\Pi}).$$

Subtracting the definitions of A_ℓ and $\hat{A}_{\mathrm{che}}$ gives

$$A_\ell - \hat{A}_{\mathrm{che}} = (\Pi A \Pi - \hat{\Pi} A \hat{\Pi}) + (\alpha-\mu)(\hat{\Pi} - \Pi).$$

Using

$$\Pi A \Pi - \hat{\Pi} A \hat{\Pi} = (\Pi - \hat{\Pi}) A \Pi + \hat{\Pi} A (\Pi - \hat{\Pi})$$

together with $\|\Pi\|_2 = \|\hat{\Pi}\|_2 = 1$, we obtain

$$\|A_\ell - \hat{A}_{\mathrm{che}}\|_2 \leq \big(2\|A\|_2 + |\alpha-\mu|\big)\|\Pi - \hat{\Pi}\|_2.$$

Now set $X = A_\ell + \mu I$ and $Y = P_{\mathrm{che}}^{(A_\mu)}$. Since $A_\ell \succeq 0$ and $\hat{A}_{\mathrm{che}} \succeq 0$, both X and Y are bounded below by μI. By the resolvent identity,

$$X^{-1} - Y^{-1} = X^{-1}(Y - X)Y^{-1},$$

and noting that $X - Y = (A_\ell + \mu I) - P_{\mathrm{che}}^{(A_\mu)} = A_\ell - \hat{A}_{\mathrm{che}}$, we have

$$\|X^{-1} - Y^{-1}\|_2 \leq \frac{1}{\mu^2}\|A_\ell - \hat{A}_{\mathrm{che}}\|_2 \leq \frac{2\|A\|_2 + |\alpha-\mu|}{\mu^2}\|\Pi - \hat{\Pi}\|_2. \tag{22}$$

Since $\|\Pi - \hat{\Pi}\|_2 = \sin\Theta(\mathrm{range}(U_\ell), \mathrm{range}(\hat{U}_\ell)) \leq \delta$ by Remark 8, we obtain

$$\|(A_\ell+\mu I)^{-1} - (P_{\mathrm{che}}^{(A_\mu)})^{-1}\|_2 \leq \frac{2\|A\|_2 + |\alpha-\mu|}{\mu^2}\delta. \tag{23}$$

We next estimate the first term. In the eigenbasis of A, the operator $(A+\mu I)^{-1}$ is diagonal with eigenvalues $(\lambda_i+\mu)^{-1}$. On the other hand,

$$(A_\ell+\mu I)^{-1} = U_\ell(\Lambda_\ell + \mu I_\ell)^{-1}U_\ell^\top + \alpha^{-1}(I - U_\ell U_\ell^\top).$$

Thus $(A_\ell+\mu I)^{-1}$ agrees with $(A+\mu I)^{-1}$ on the captured upper and lower spectral blocks, while on the orthogonal complement it acts as the scalar α^{-1}. Therefore

$$\|(A+\mu I)^{-1} - (A_\ell+\mu I)^{-1}\|_2 = \max_{i\in\mathcal{I}_{\mathrm{mid}}}\left|\frac{1}{\lambda_i+\mu} - \frac{1}{\alpha}\right|. \tag{24}$$

Finally, combining (23) and (24) with the initial triangle inequality, the estimate (21) follows. □

Theorem 12 says that, relative to the oracle model, the practical preconditioner incurs only the projector perturbation term. When the filtered randomized extraction captures the extremal eigenspaces accurately and thus $\tan\Theta(\mathrm{range}(U_\ell), \mathrm{range}(\hat{U}_\ell))$ can be bounded by a small δ, this term is small, so the practical preconditioner preserves the spectral improvement predicted by the oracle model.

We also point out that the above argument extends without difficulty to the unshifted case when the original matrix is strictly positive definite. The assumption $\mu > 0$ is used in (22) only to provide a uniform lower bound for the two resolvents.

Remark 13 If $A \succ 0$, then the assumption $\mu > 0$ in Theorem 12 can be relaxed to $\mu \geq 0$, since the positive definiteness of A provides the required uniform lower bound. More precisely, set

$$\gamma_\mu := \min\{\lambda_{\min}(A) + \mu, \alpha\}.$$

Since $A \succeq \lambda_{\min}(A)I$, the Ritz matrix $\hat{U}_\ell^\top A \hat{U}_\ell$ is also bounded below by $\lambda_{\min}(A)I$. Hence $X = A_\ell + \mu I \succeq \gamma_\mu I$ and $Y = P_{\mathrm{che}}^{(A_\mu)} \succeq \gamma_\mu I$. Consequently, the resolvent identity yields the refined estimate

$$\|X^{-1} - Y^{-1}\|_2 \leq \frac{1}{\gamma_\mu^2}\|A_\ell - \hat{A}_{\mathrm{che}}\|_2 \leq \frac{2\|A\|_2 + |\alpha - \mu|}{\gamma_\mu^2}\|\Pi - \hat{\Pi}\|_2. \tag{25}$$

In particular, when $A \succ 0$ and $\mu = 0$, this gives

$$\|X^{-1} - Y^{-1}\|_2 \leq \frac{2\|A\|_2 + \alpha}{\min\{\lambda_{\min}(A), \alpha\}^2}\|\Pi - \hat{\Pi}\|_2. \tag{26}$$

Thus, in the strictly positive definite case, the regularization parameter μ may be set to zero; the denominator in the perturbation bound is then controlled by the smaller of the smallest eigenvalue $\lambda_{\min}(A)$ and the scalar parameter α.

The following result gives the main condition-number estimate for the practical preconditioner. The factor b_μ/a_μ is inherited from the oracle model and reflects the quality of the scalar approximation on the uncaptured middle spectrum, while the factor $(1+\tau)/(1-\tau)$ quantifies the perturbation caused by replacing the exact extremal eigenspace by its Ritz approximation.

Theorem 14 *Let $A_\mu = A + \mu I \succ 0$, $P_*^{(A_\mu)} = U_\ell(\Lambda_\ell + \mu I_\ell)U_\ell^\top + \alpha(I - \Pi)$, and $P_{\mathrm{che}}^{(A_\mu)} = \hat{U}_\ell(\hat{\Lambda}_\ell + \mu I_\ell)\hat{U}_\ell^\top + \alpha(I - \hat{\Pi})$ where $\Pi = U_\ell U_\ell^\top$, $\hat{\Pi} = \hat{U}_\ell \hat{U}_\ell^\top$. Assume that $\lambda_i + \mu \in [a_\mu, b_\mu]$ for all $i \notin \mathcal{J}$, where $\mathcal{J} = \{1, \ldots, \ell_1\} \cup \{n - \ell_2 + 1, \ldots, n\}$ and $\alpha \in [a_\mu, b_\mu]$. Define $s = \|\Pi - \hat{\Pi}\|_2$, $d_{\min} = \min_{j \in \mathcal{J}}(\lambda_j + \mu)$, and $\gamma = \min\{\alpha, d_{\min}\}$. Let $\tau = \frac{\alpha + 2\|A_\mu\|_2}{\gamma} s$. If $\tau < 1$, then*

$$\kappa_2\Big(\big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2}\Big) \leq \frac{1+\tau}{1-\tau} \cdot \frac{b_\mu}{a_\mu}. \tag{27}$$

Proof We first compare the practical preconditioner with the oracle one. Set

$$M = \big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2}, \quad M_* = \big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2},$$
$$S = \big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2} \big(P_*^{(A_\mu)}\big)^{1/2}.$$

Since $A_\mu = \big(P_*^{(A_\mu)}\big)^{1/2} M_* \big(P_*^{(A_\mu)}\big)^{1/2}$, we have $M = S M_* S^\top$. Therefore,

$$\kappa_2(M) \le \kappa_2(S)\kappa_2(M_*)\kappa_2(S^\top) = \kappa_2(S)^2 \kappa_2(M_*). \tag{28}$$

Moreover,

$$S^\top S = \big(P_*^{(A_\mu)}\big)^{1/2} \big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1} \big(P_*^{(A_\mu)}\big)^{1/2}.$$

This matrix is similar to $\big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1} P_*^{(A_\mu)}$, whose condition number is the same as that of $P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1}$. Therefore, it holds that $\kappa_2(S)^2 = \kappa_2\Big(P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1}\Big)$. Hence, it follows from (28) that

$$\begin{aligned} \kappa_2\Big(\big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_{\mathrm{che}}^{(A_\mu)}\big)^{-1/2}\Big) &\le \kappa_2\Big(P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1}\Big) \\ &\cdot \kappa_2\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2}\Big). \end{aligned} \tag{29}$$

We next bound the two factors on the right-hand side. Since $P_*^{(A_\mu)}$ is the oracle subspace-complement preconditioner and $\alpha \in [a_\mu, b_\mu]$, it follows from (18) that

$$\kappa_2\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} A_\mu \big(P_*^{(A_\mu)}\big)^{-1/2}\Big) \le \frac{b_\mu}{a_\mu}. \tag{30}$$

To estimate the perturbation factor $\kappa_2\Big(P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1}\Big)$, we first note that $P_{\mathrm{che}}^{(A_\mu)} (P_*^{(A_\mu)})^{-1}$ is similar to the symmetric matrix $\big(P_*^{(A_\mu)}\big)^{-1/2} P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1/2}$. Define

$$E = \big(P_*^{(A_\mu)}\big)^{-1/2} \Big(P_{\mathrm{che}}^{(A_\mu)} - P_*^{(A_\mu)}\Big) \big(P_*^{(A_\mu)}\big)^{-1/2}.$$

Then

$$\big(P_*^{(A_\mu)}\big)^{-1/2} P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1/2} = I + E.$$

It remains to estimate the two factors $\Big\|\big(P_*^{(A_\mu)}\big)^{-1}\Big\|_2$ and $\Big\|P_{\mathrm{che}}^{(A_\mu)} - P_*^{(A_\mu)}\Big\|_2$. First, since the spectrum of $P_*^{(A_\mu)}$ consists of α on $\mathrm{range}(U_\ell)^\perp$ and $\{\lambda_j + \mu : j \in \mathcal{J}\}$ on $\mathrm{range}(U_\ell)$, we have $\lambda_{\min}\Big(P_*^{(A_\mu)}\Big) = \gamma$ and $\Big\|\big(P_*^{(A_\mu)}\big)^{-1}\Big\|_2 = \frac{1}{\gamma}$. Next, using

$$P_*^{(A_\mu)} = \alpha(I - \Pi) + \Pi A_\mu \Pi, \qquad P_{\mathrm{che}}^{(A_\mu)} = \alpha(I - \widehat{\Pi}) + \widehat{\Pi} A_\mu \widehat{\Pi},$$

we obtain

$$P_{\mathrm{che}}^{(A_\mu)} - P_*^{(A_\mu)} = \alpha(\Pi - \widehat{\Pi}) + \Big(\widehat{\Pi} A_\mu \widehat{\Pi} - \Pi A_\mu \Pi\Big).$$

Moreover,

$$\widehat{\Pi} A_\mu \widehat{\Pi} - \Pi A_\mu \Pi = (\widehat{\Pi} - \Pi) A_\mu \widehat{\Pi} + \Pi A_\mu (\widehat{\Pi} - \Pi).$$

Together with $\|\Pi\|_2 = \|\widehat{\Pi}\|_2 = 1$, this gives

$$\Big\|P_{\mathrm{che}}^{(A_\mu)} - P_*^{(A_\mu)}\Big\|_2 \le (\alpha + 2\|A_\mu\|_2)\, s.$$

Therefore,

$$\|E\|_2 \le \Big\|\big(P_*^{(A_\mu)}\big)^{-1}\Big\|_2 \Big\|P_{\mathrm{che}}^{(A_\mu)} - P_*^{(A_\mu)}\Big\|_2 \le \frac{\alpha + 2\|A_\mu\|_2}{\gamma}\, s = \tau.$$

By assumption, $\tau < 1$. Since E is symmetric, this implies

$$(1 - \tau) I \preceq I + E \preceq (1 + \tau) I.$$

Hence

$$\kappa_2\Big(\big(P_*^{(A_\mu)}\big)^{-1/2} P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1/2}\Big) \le \frac{1 + \tau}{1 - \tau}.$$

Consequently,

$$\kappa_2\Big(P_{\mathrm{che}}^{(A_\mu)} \big(P_*^{(A_\mu)}\big)^{-1}\Big) \le \frac{1 + \tau}{1 - \tau}. \tag{31}$$

Finally, combining (29), (30), and (31) yields (27). □

Remark 15 The requirement $\tau < 1$ is mild in the regime relevant to our construction. To see this, recall that

$$\tau = \frac{\alpha + 2\|A_\mu\|_2}{\gamma}\, s, \qquad s = \|\Pi - \hat{\Pi}\|_2.$$

By Remark 8,

$$s = \|\Pi - \hat{\Pi}\|_2 = \sin\Theta(\mathrm{range}(U_\ell), \mathrm{range}(\hat{U}_\ell)) \le \delta,$$

where δ is controlled by the subspace approximation error of the Chebyshev-filtered sketch. In particular, when the target spectrum is well separated from the bulk spectrum, the estimate in Theorem 4 shows that this error decreases rapidly with the polynomial degree. Consequently, the subspace mismatch s can be made sufficiently small, and hence the condition $\tau < 1$ is typically attainable in practice.

To illustrate this point more clearly, we consider a synthetic example based on the construction in Section 6.1, with $\lambda_{\mathrm{bulk}} \subset [0.8, 1.2]$, $\lambda_{\mathrm{small}} \subset [10^{-4}, 10^{-2}]$, and $\lambda_{\mathrm{large}} \subset [2, 10]$. We vary the Chebyshev polynomial degree over

$$m \in \{0, 2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100\}.$$

Figure 1 reports the corresponding values of τ and s. As the degree m increases, the subspace mismatch s decreases rapidly, and consequently τ drops below the threshold $\tau = 1$ around $m = 30$. This supports the claim that the condition $\tau < 1$ is attainable in the regime where the Chebyshev-filtered sketch accurately captures the target invariant subspace.

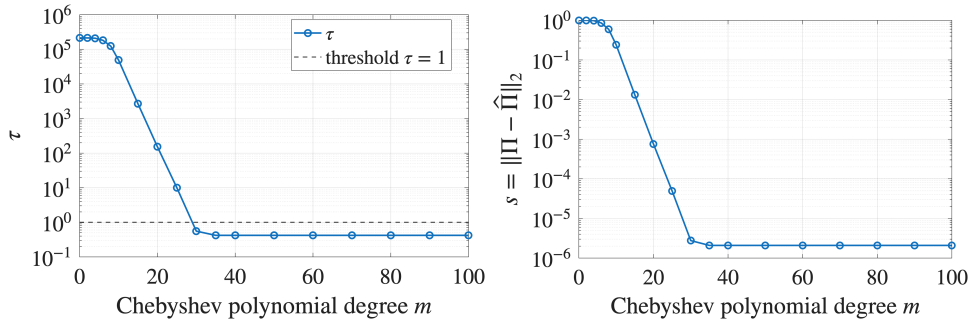


Fig. 1: Dependence on the Chebyshev polynomial degree m. Panel (a) shows the parameter τ; panel (b) shows the subspace mismatch s.

5 Implementation

The preceding sections describe the structure of the proposed low-rank preconditioner once suitable extremal spectral information is available. In practice, an important issue is the selection of a suitable spectral interval for the Chebyshev filter. When the spectrum is wide or asymmetric, applying the Chebyshev filter directly to

A_μ can be sensitive to the prescribed spectral interval. To make the filtering step more robust in practice, we adopt a two-stage strategy. We first apply an inexpensive base preconditioner to reshape the spectrum, and then construct the low-rank correction by randomized subspace extraction and Chebyshev filtering.

5.1 Spectrum shaping by a base preconditioner

The base preconditioner is used mainly to reshape the spectrum. It reduces the main spectral spread and makes the subsequent Chebyshev interval selection more predictable. Let $M_0 \in \mathbb{R}^{n\times n}$ be a symmetric positive definite base preconditioner for A_μ. We write

$$M_0 = R^\top R,$$

where R is nonsingular, and define the spectrally shaped operator

$$B := R^{-T} A_\mu R^{-1}.$$

For the implementation considered in this work, we use Jacobi scaling as the base preconditioner. Let $D = \mathrm{diag}(A_\mu)$. Then $M_0 = D$, $R = D^{1/2}$, and

$$B = D^{-1/2} A_\mu D^{-1/2}.$$

For instance, when A_μ is symmetric diagonally dominant with positive diagonal, Jacobi scaling places the shaped spectrum in $[0, 2]$ by the Gershgorin theorem.

5.2 Practical interval selection

After the spectrum-shaping step, the remaining practical issue is the choice of the interval for the Chebyshev filter. A high-degree Chebyshev filter can be numerically unsafe when it is applied to an upper outlier on a wide spectral scale. For example, if the non-target interval is $[0, 1.2]$ and an upper outlier is located at $\lambda = 100$, then $\phi(100) \approx 165.7$, and hence $|T_m(\phi(100))| \approx \frac{1}{2}(331.3)^m$, which reaches the double-precision overflow scale at about $m = 122$. Thus, increasing the Chebyshev degree on the upper tail is not a robust strategy when the spectral scale is wide.

For this reason, we treat the two spectral tails differently in the practical implementation. The upper tail is first captured by a standard randomized extraction and is used to determine the spectral scale of the shaped operator. This scale estimate is essential for choosing a safe right endpoint of the Chebyshev filtering interval. The lower tail is then extracted by Chebyshev-filtered randomized sketching.

Let ℓ_1 be the dimension of the upper-tail pilot space. We draw $\Omega_{\ell_1} \in \mathbb{R}^{n\times \ell_1}$, form $Y_{\ell_1} = B\Omega_{\ell_1}$, compute $Q_{\ell_1} = \mathrm{orth}(Y_{\ell_1})$, and set $T_{\ell_1} = Q_{\ell_1}^\top B Q_{\ell_1}$. After the eigendecomposition $T_{\ell_1} = W_{\ell_1} \hat{\Theta}_{\ell_1} W_{\ell_1}^\top$, we set $\hat{U}_{\ell_1} = Q_{\ell_1} W_{\ell_1}$. The largest Ritz value $\hat{\lambda}_{\max} := \lambda_{\max}(T_{\ell_1})$ is then used as the spectral scale of B for choosing the right endpoint of the Chebyshev interval.

With this upper spectral scale in hand, we now choose the Chebyshev interval for the subsequent lower-tail extraction. Specifically, we set its right endpoint by

$$b = \gamma \hat{\lambda}_{\max}, \tag{32}$$

where $\gamma > 1$ is a safety factor. The purpose of this choice is to make the subsequent lower-tail extraction more robust. By setting the right endpoint slightly above the estimated spectral scale, the bulk and large eigenvalues are kept inside the bounded region of the Chebyshev polynomial, while the small eigenvalues below the left endpoint are amplified. In our experiments we use $\gamma = 2$. The choice of the left endpoint a is discussed in Section 6.1.4. Once $[a, b]$ has been fixed, we apply Algorithm 1 to B in order to extract the lower spectral block.

The two-stage Chebyshev preconditioner is summarized in Algorithm 2.

Algorithm 2 Two-stage Chebyshev preconditioner

Require: SPD matrix $A_\mu = A + \mu I$, base preconditioner $M_0 = R^\top R$, extraction dimensions ℓ_1, ℓ_2, Chebyshev degree m, left endpoint $a > 0$, safety factor $\gamma > 1$.
Ensure: Preconditioner model $\mathcal{M} := (R, \hat{U}_\ell, \hat{\Theta}_\ell, \alpha)$.

1: Form the shaped operator $B = R^{-T} A_\mu R^{-1}$
2: Draw $\Omega_{\ell_1} \in \mathbb{R}^{n \times \ell_1}$ and form $Y_{\ell_1} = B\Omega_{\ell_1}$, $Q_{\ell_1} = \mathrm{orth}(Y_{\ell_1})$, and $T_{\ell_1} = Q_{\ell_1}^\top B Q_{\ell_1}$
3: Compute $T_{\ell_1} = W_{\ell_1} \hat{\Theta}_{\ell_1} W_{\ell_1}^\top$, set $\hat{U}_{\ell_1} = Q_{\ell_1} W_{\ell_1}$, and let $\hat{\lambda}_{\max} = \lambda_{\max}(T_{\ell_1})$
4: Set the right endpoint $b = \gamma \hat{\lambda}_{\max}$
5: Apply the Chebyshev-filtered randomized extraction to B on $[a, b]$ with target rank ℓ_2, and degree m to obtain $(\hat{U}_{\ell_2}, \hat{\Theta}_{\ell_2})$, where $\hat{U}_{\ell_2} \in \mathbb{R}^{n \times \ell_2}$
6: Set $\hat{a}_{\mathrm{mid}} = \lambda_{\max}(\hat{\Theta}_{\ell_2})$ and $\hat{b}_{\mathrm{mid}} = \lambda_{\min}(\hat{\Theta}_{\ell_1})$, and choose $\alpha = \sqrt{\hat{a}_{\mathrm{mid}} \hat{b}_{\mathrm{mid}}}$
7: Form $Q_\ell = \mathrm{orth}\big([\hat{U}_{\ell_1}, \hat{U}_{\ell_2}]\big)$ and $T_\ell = Q_\ell^\top B Q_\ell$
8: Compute $T_\ell = W_\ell \hat{\Theta}_\ell W_\ell^\top$ and set $\hat{U}_\ell = Q_\ell W_\ell$
9: Return $\mathcal{M} := (R, \hat{U}_\ell, \hat{\Theta}_\ell, \alpha)$

Given the output $\mathcal{M} = (R, \hat{U}_\ell, \hat{\Theta}_\ell, \alpha)$ of Algorithm 2, we construct the subspace-complement preconditioner on the shaped operator:

$$\tilde{P} = \hat{U}_\ell \hat{\Theta}_\ell \hat{U}_\ell^\top + \alpha\big(I - \hat{U}_\ell \hat{U}_\ell^\top\big). \tag{33}$$

For the Jacobi base preconditioner, let $D = \mathrm{diag}(A_\mu)$, so that $R = D^{1/2}$. Returning to the original variables, the final preconditioner is

$$P = D^{1/2} \tilde{P} D^{1/2}. \tag{34}$$

Algorithm 3 Chebyshev-filtered PCG

Require: SPD matrix $A_\mu = A + \mu I$, right-hand side b, initial guess x_0, and preconditioner model $\mathcal{M} = (D, \hat{U}_\ell, \hat{\Theta}_\ell, \alpha)$ constructed by Algorithm 2.
Ensure: Approximate solution x to $A_\mu x = b$.

Define the preconditioner action $F(r) = P^{-1}r$ by (35).
$r_0 = b - A_\mu x_0$
$z_0 = F(r_0)$
$p_0 = z_0$
$\rho_0 = r_0^\top z_0$
for $j = 0, 1, 2 \ldots$ **do**
    $v_j = A_\mu p_j$
    $\omega_j = \rho_j / (p_j^\top v_j)$
    $x_{j+1} = x_j + \omega_j p_j$
    $r_{j+1} = r_j - \omega_j v_j$
    $z_{j+1} = F(r_{j+1})$
    $\rho_{j+1} = r_{j+1}^\top z_{j+1}$
    $\beta_j = \rho_{j+1} / \rho_j$
    $p_{j+1} = z_{j+1} + \beta_j p_j$
    if a stopping criterion is satisfied, stop and return x_{j+1}
end for

5.3 Use within PCG

In the iterative phase, the preconditioner is used only through the action of its inverse. Given the Jacobi-scaled model

$$\mathcal{M} = (D, \hat{U}_\ell, \hat{\Theta}_\ell, \alpha),$$

we define the preconditioner handle

$$F(r) = D^{-1/2}\left[\hat{U}_\ell \hat{\Theta}_\ell^{-1} \hat{U}_\ell^\top (D^{-1/2} r) + \alpha^{-1}\big(I - \hat{U}_\ell \hat{U}_\ell^\top\big)(D^{-1/2} r)\right]. \tag{35}$$

Thus $F(r) = P^{-1}r$, and this handle can be supplied directly to a standard PCG routine. The solve phase is summarized in Algorithm 3.

6 Numerical experiments

In this section, we evaluate the effectiveness of the proposed preconditioner and its impact on PCG convergence on synthetic and real data. All numerical experiments were implemented in MATLAB R2025b on a machine equipped with an Apple M5 Pro chip and 48 GB of unified memory. The code to reproduce our results can be found at https://github.com/dongzeyu67/chebyshev-randomized-preconditioner.

In Section 6.1, we first use diagnostic experiments to examine the effect of Chebyshev filtering on the two spectral tails. In these experiments, both the upper and lower tails are treated by Chebyshev-filtered randomized extraction, demonstrating

that polynomial filtering can recover the relevant extreme eigenspaces and improve the quality of the resulting low-rank correction. As shown in Section 6.1.3, applying Chebyshev filtering to the upper tail over a wide spectral range can lead to severe finite-precision instability caused by excessive polynomial amplification. Therefore, in the PCG experiments in Sections 6.2 and 6.3, we use the two-stage construction described in Algorithm 2. The upper spectral tail is captured by a standard randomized range finder, whereas the lower spectral tail is extracted by a Chebyshev-filtered randomized sketch.

6.1 Preconditioner quality on synthetic spectra

We first assess the quality of the proposed preconditioner on synthetic SPD matrices with prescribed spectra. Throughout this subsection, we construct $A = U\,\mathrm{diag}(\lambda_1, \ldots, \lambda_n)U^\top$, where $U \in \mathbb{R}^{n\times n}$ is a random orthogonal matrix and $n = 1000$. The spectrum consists of three parts: a bulk cluster in $[0.8, 1.2]$, $n_{\mathrm{small}} = 30$ small eigenvalues distributed logarithmically in $[10^{-12}, 10^{-6}]$, and $n_{\mathrm{large}} = 5$ large eigenvalues distributed logarithmically in a prescribed interval. Thus, the matrix has a well-conditioned middle spectrum near one together with a small number of isolated outliers at both spectral ends. This construction matches the intended use of the method: the preconditioner corrects the difficult extremal modes explicitly, while representing the remaining bulk spectrum by a scalar approximation.

The intervals used in the two Chebyshev filters are chosen directly from the prescribed spectral blocks. For the upper-tail extraction, we take $[a_L, b_L] = [0, \lambda_{\mathrm{bulk,max}}]$, so that the lower tail and the bulk lie inside the bounded region of the polynomial while the large eigenvalues lie outside and are amplified. For the lower-tail extraction, we take $[a_S, b_S] = [\lambda_{\mathrm{bulk,min}},\ \lambda_{\mathrm{large,max}}]$, so that the bulk and the upper tail are included in the non-target interval, while the small eigenvalues remain outside on the left and are amplified by the filter.

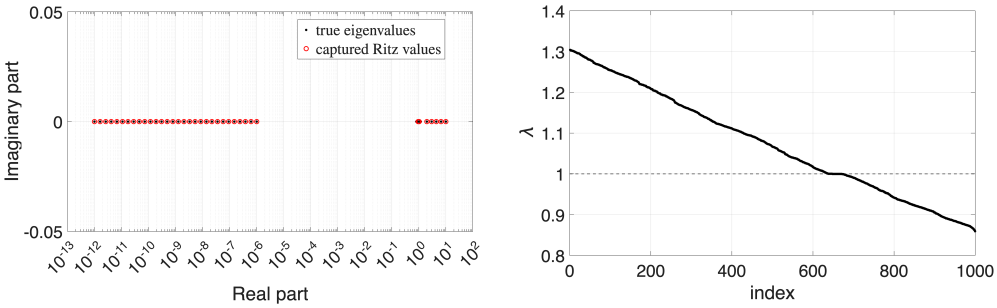
To study the efficiency of a preconditioner P, we use three quantities: $\eta_{A^{-1}}(P) = \|P^{-1} - A^{-1}\|_2 / \|A^{-1}\|_2$, $\eta_{\mathrm{spec}}(P) = \|P^{-1/2}AP^{-1/2} - I\|_2$ and $\kappa_P = \kappa\big(P^{-1/2}AP^{-1/2}\big)$. The first quantity measures the relative error in approximating A^{-1}, and is therefore particularly sensitive to the treatment of the smallest eigenvalues. The second measures the deviation of the preconditioned spectrum from the ideal value 1 and the third is the condition number directly relevant to PCG convergence.

6.1.1 Feasibility

We first consider a synthetic example with a moderately separated upper spectral block, taking $\lambda_{\mathrm{bulk}} \subset [0.8, 1.2]$, $\lambda_{\mathrm{small}} \subset [10^{-12}, 10^{-6}]$, and $\lambda_{\mathrm{large}} \subset [2, 10]$. The purpose of this test is to examine whether the proposed Chebyshev preconditioner can capture both spectral tails and produce a meaningful spectral correction when the upper range is of moderate width. The Chebyshev degrees for the large- and small-eigenvalue tails are denoted by m_L and m_S, respectively, and are set to $m_L = 10$ and $m_S = 40$. The target ranks are chosen as $\ell_1 = n_{\mathrm{large}} + 10$ and $\ell_2 = n_{\mathrm{small}} + 10$.

Figure 2 shows the spectral behavior of the proposed preconditioner. In panel (a), the computed Ritz values track the extremal eigenvalues of A closely, indicating that

the extracted subspaces capture the relevant upper and lower spectral components with reasonable accuracy. In panel (b), the spectrum of $P^{-1/2}AP^{-1/2}$ is concentrated near 1; the eigenvalues lie roughly in the interval $[0.86, 1.33]$, which is substantially narrower than the original spectral range. These observations show that the proposed preconditioner is effective and produces a clear clustering effect on the preconditioned spectrum.



(a) Spectrum of A and captured Ritz values.

(b) Spectrum of $P^{-1/2}AP^{-1/2}$.

Fig. 2: Panel (a) compares the eigenvalues of A with the captured Ritz values; panel (b) shows the spectrum of $P^{-1/2}AP^{-1/2}$.

6.1.2 Effects of rank and degree

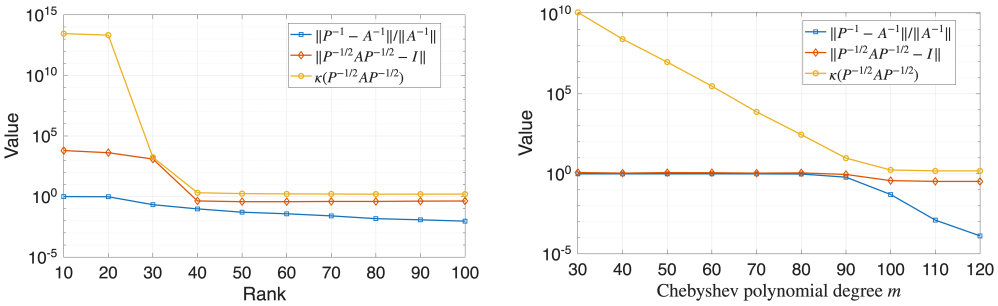
We next consider a more demanding synthetic setting in which the upper spectral block is widened from $[2, 10]$ to $[10, 100]$, while all other aspects of the construction are kept unchanged. In particular, the bulk eigenvalues remain in $[0.8, 1.2]$, the small eigenvalues remain in $[10^{-12}, 10^{-6}]$, and the same interval-selection rules are used for the two Chebyshev filters. Thus, the upper-tail filter is built on $[a_L, b_L] = [0, 1.2]$, whereas the lower-tail filter is built on $[a_S, b_S] = [0.8, 100]$. Since the right endpoint of the lower-tail interval now scales with the enlarged upper spectral range, the corresponding Chebyshev filter becomes substantially less selective at low degrees. This regime therefore provides a useful test of how the method depends on the rank parameter and on the polynomial degree.

To isolate the effect of the rank, we first fix the polynomial degrees at $m_L = 10$ and $m_S = 100$, and vary the target rank from 10 to 100 in steps of 10. The results are shown in Figure 3(a). As the rank increases, the inverse error $\eta_{A^{-1}}(P)$ decreases steadily, with the most substantial improvement occurring once the rank reaches 30. This behavior is consistent with the construction of the test matrix: the lower spectral block contains 30 small eigenvalues, so the low-rank correction cannot be expected to resolve the near-null subspace before the assigned rank reaches this intrinsic dimension. Once the rank is large enough to represent this block, the approximation improves markedly. Beyond this point, the gain becomes much smaller and the curve essentially levels off. The spectral error $\eta_{\mathrm{spec}}(P)$ and the condition number κ_P show the

same transition, indicating that the preconditioner becomes effective after the relevant extremal subspace is sufficiently captured.

We then fix the rank at 50 and study the influence of the lower-tail Chebyshev degree m_S. The upper-tail degree m_L is kept fixed because the numerical instability caused by taking m_L excessively large is a separate issue and will be discussed in Section 6.1.3. Specifically, we keep $m_L = 10$ fixed and vary m_S from 30 to 120 in steps of 10. The results are shown in Figure 3(b). For moderate values of m_S, both error measures remain nearly unchanged, indicating that the lower-tail filter is still too weak to separate the near-null components from the wide bulk interval. Once m_S is large enough, the inverse error drops by several orders of magnitude. This reflects the wider lower-tail filtering interval, which requires a substantially larger polynomial degree before the near-null components can be effectively separated.

Taken together, Figure 3 shows that widening the upper spectral block does not invalidate the Chebyshev-based extremal correction strategy, but it makes the lower-tail extraction significantly more demanding. In this regime, both a sufficiently large rank and a sufficiently high lower-tail polynomial degree are necessary before the preconditioner begins to approximate A^{-1} accurately and to compress the preconditioned spectrum.



(a) Influence of the rank parameter.

(b) Influence of the Chebyshev degree m_S.

Fig. 3: Widened upper spectral block $\lambda_{\text{large}} \subset [10, 100]$. Panel (a) shows the effect of the target rank; panel (b) shows the effect of the lower-tail Chebyshev degree m_S.

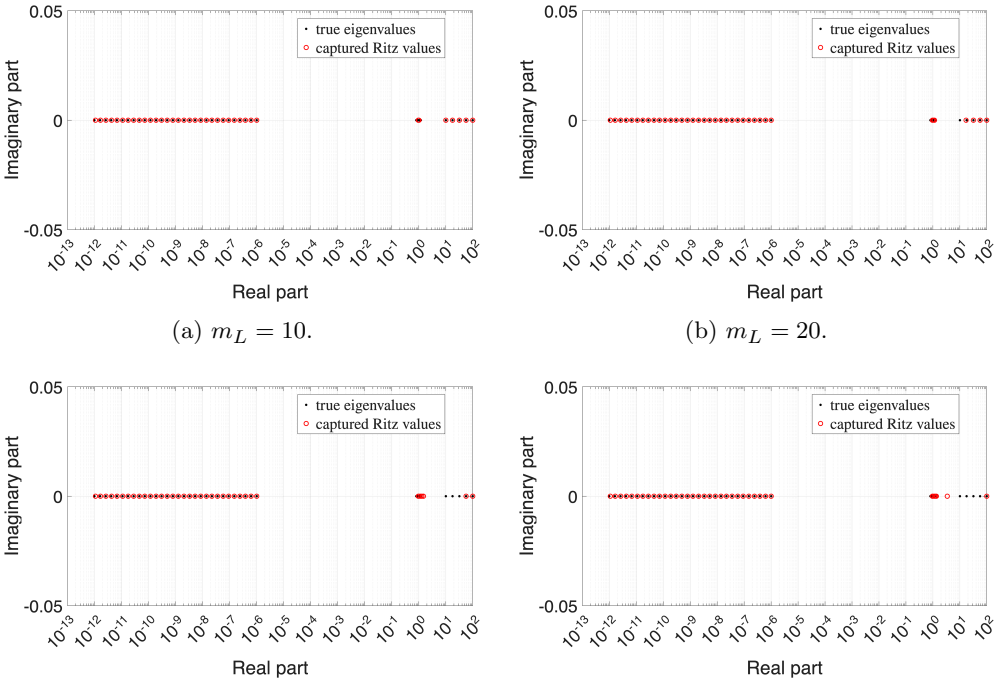
6.1.3 High-degree instability of upper-tail filtering

We next examine the effect of the Chebyshev degree on the upper spectral block. In this experiment, we keep the target rank fixed at 50, retain the same spectral intervals and all other parameters as above, and fix the lower-tail degree at $m_S = 100$. We then vary the upper-tail degree m_L over the values $10, 20, 50$, and 100. The corresponding results are shown in Figure 4.

Figure 4 shows that increasing the upper-tail degree does not improve the extraction of the large-eigenvalue subspace; in fact, the results become progressively worse as m_L increases. When $m_L = 10$, the dominant eigenvalues are still captured reasonably well. As m_L increases to 20, 50, and 100, however, the extracted Ritz values

deteriorate and drift away from the true dominant eigenvalues. In the most extreme case $m_L = 100$, the upper-tail extraction is no longer reliable. This behavior indicates that, when the upper spectral range is wide, a high-degree Chebyshev filter on the large-eigenvalue side may suffer from severe floating-point amplification. Since the polynomial grows very rapidly outside the reference interval, the three-term recurrence becomes increasingly sensitive to roundoff errors, and the intended spectral selectivity can be degraded.

This observation is important for the practical design of the method. It shows that, for a wide upper spectral range, increasing the upper-tail Chebyshev degree is not a robust strategy. In our implementation, this motivates a two-stage design with different treatments of the upper and lower spectral tails: a cheap base preconditioner is first used to compress the spectral range, the upper spectral block is then extracted by a standard randomized subspace procedure, and the lower spectral block is treated by Chebyshev-filtered randomized extraction.



(a) $m_L = 10$. (b) $m_L = 20$.

(c) $m_L = 50$. (d) $m_L = 100$.

Fig. 4: Effect of the upper-tail Chebyshev degree for the widened upper spectral block $\lambda_{\text{large}} \subset [10, 100]$. Panels (a)–(d) correspond to $m_L = 10, 20, 50, 100$, respectively.

6.1.4 Choice of the left endpoint

In this subsection, we comment on the choice of the left endpoint in the lower-tail Chebyshev interval. The right endpoint is relatively easy to choose in the two-stage implementation, since the pilot upper-tail computation gives the scale estimate used in (32), namely $b = 2\hat{\lambda}_{\max}$. The left endpoint a controls how much of the lower spectrum is placed outside the filtering interval and therefore treated as the target region. A smaller a gives a narrower target region and may miss some near-null modes, whereas a larger a requires a larger retained lower-tail rank.

This point can be illustrated with a simple example based on the one-dimensional Dirichlet Laplacian. Let $A = h^{-2}\,\mathrm{tridiag}(-1, 2, -1)$, with $h = 1/(n+1)$, and let $B = D^{-1/2}AD^{-1/2}$, where $D = \mathrm{diag}(A)$. Then B has eigenvalues $\theta_j(B) = 2\sin^2(j\pi/(2(n+1)))$, for $j = 1, \ldots, n$. For a prescribed lower-tail rank ℓ_2, a natural rank-consistent choice of the left endpoint is

$$a_{\ell_2} = \frac{\theta_{\ell_2}(B) + \theta_{\ell_2+1}(B)}{2}.$$

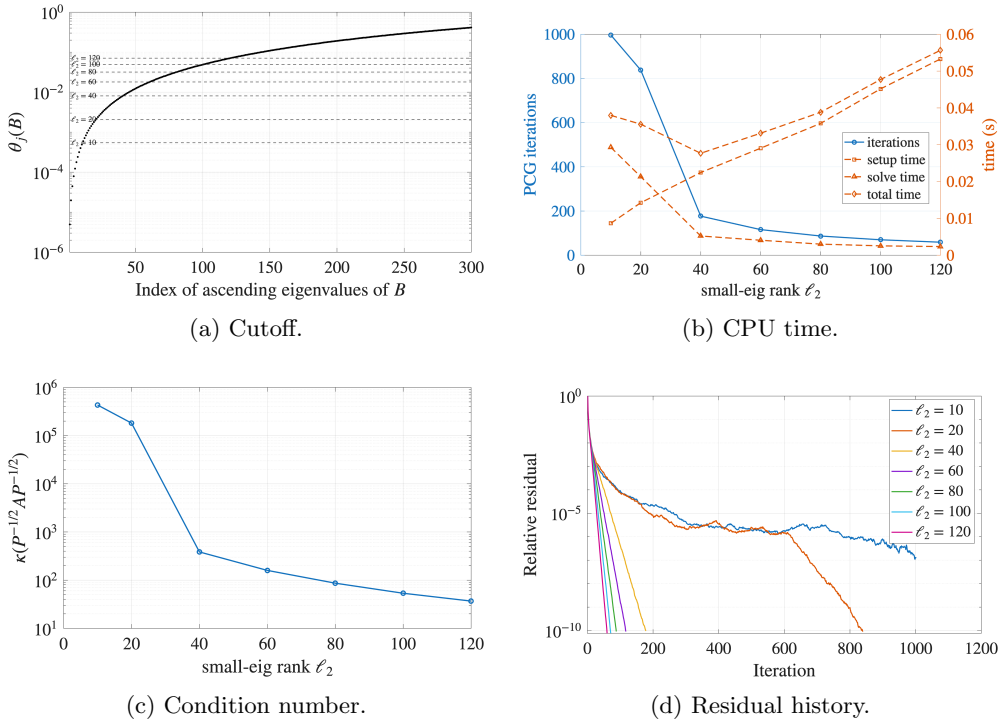
With this choice, exactly ℓ_2 eigenvalues lie below the left endpoint and are therefore placed in the amplified part of the Chebyshev filter. Consequently, moving a to the right is equivalent to asking the preconditioner to capture more low-end modes.

Figure 5 illustrates this behavior for the Jacobi-scaled Laplacian. Small values of ℓ_2, and hence small left endpoints, leave some important low-frequency components untreated and lead to slow PCG convergence. Increasing ℓ_2 improves the effective smallest eigenvalue and reduces the iteration count. For a single linear solve, beyond a moderate rank, the additional reduction in solve time may be offset by the increasing setup cost. In applications involving multiple right-hand sides or time-dependent problems, however, the setup cost may be amortized over many related solves, and therefore larger ranks can still be advantageous. This behavior illustrates the practical coupling between the left endpoint and the prescribed lower-tail rank: moving the cutoff to the right can improve the correction of low-end modes, but it also requires a larger extracted subspace. In the subsequent experiments, where the detailed spectral distributions are not prescribed, the retained ranks and the left endpoint are treated as practical algorithmic choices; we use the simple value $a = 0.1$ as the lower-tail cutoff.

The preceding discussion is specific to the Jacobi-scaled operator. Since Jacobi scaling does not remove the mesh dependence of the lower spectrum, the appropriate left endpoint and retained lower-tail rank may still vary with the discretization. With a stronger base preconditioner, the shaped spectrum may contain a more stable bulk and fewer persistent outliers, leading to different endpoint choices. Adaptive, problem-dependent endpoint selection is left for future work.

6.2 Experiments on synthetic data

We now turn to the actual linear solver and study the effect of the proposed preconditioner on PCG convergence. Throughout this subsection, we compare four methods for solving SPD systems: plain conjugate gradients (CG), Jacobi-preconditioned



(a) Cutoff.

(b) CPU time.

(c) Condition number.

(d) Residual history.

Fig. 5: Effect of the left endpoint for a Jacobi-scaled one-dimensional Dirichlet Laplacian. Panels (a)–(d) show the cutoff, CPU time, condition number, and residual history, respectively.

conjugate gradients (Jac-PCG), randomized Nyström-preconditioned conjugate gradients (Nys-PCG) [45], and the proposed Jacobi+Chebyshev-preconditioned conjugate gradients (Jac+Che-PCG). These experiments test whether the improvement in preconditioner quality observed in the previous subsection translates into a corresponding acceleration of the iterative solver. Unless otherwise stated, in the Jac+Che preconditioner we use polynomial degree $m = 100$, the stopping tolerance for the relative residual $\|b - Ax_k\|_2/\|b\|_2$ is set to 10^{-14} in all CG and PCG runs, and the maximum iteration count is set to n, the natural finite-termination bound for CG in exact arithmetic.

6.2.1 Synthetic matrix with exponential spectral tails

We first consider a synthetic SPD matrix whose spectrum has exponential tails at both ends. Specifically, we take $n = 2000$ and define $A = S\Lambda S^\top$, where S is a random orthogonal matrix. The smallest 30 eigenvalues are distributed exponentially from 10^{-12} to 1, the middle eigenvalues are all equal to 1, and the largest 10 eigenvalues are distributed exponentially from 1 to 10^2. The right-hand side is constructed as $b = Ax^*$, where x^* is a random vector. For Nys-PCG, the rank is chosen as $\ell_{\mathrm{Nys}} = \ell_1 + \ell_2$, so

that it uses the same total rank budget as the proposed method. In this experiment, we take $\ell_1 = 20$ and $\ell_2 = 100$.

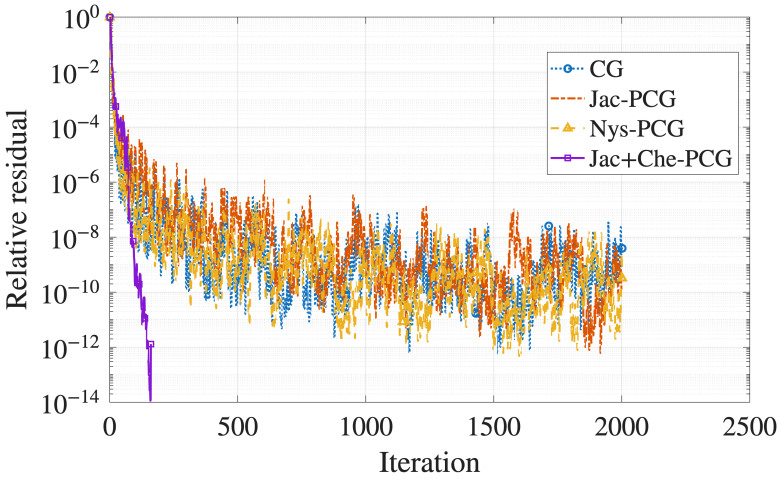


Fig. 6: Relative residual versus iteration for the synthetic matrix with exponential spectral tails.

Figure 6 shows that plain CG, Jac-PCG, and Nys-PCG all reduce the residual rapidly during the first few iterations, but then stagnate and oscillate above the prescribed tolerance. In particular, none of these three methods reaches the target tolerance within the iteration cap. By contrast, Jac+Che-PCG continues to reduce the residual and reaches the prescribed tolerance in roughly 160 iterations. This clear separation suggests that the near-null spectral block is the main difficulty in this test problem. Standard dominant-subspace correction and diagonal scaling alone do not sufficiently reduce the effect of the smallest eigenvalues on PCG convergence. By combining Jacobi scaling with a Chebyshev-filtered low-rank correction for the lower tail, the proposed method directly addresses this obstruction and therefore substantially reduces iterations.

6.2.2 A Jacobi-scaled matrix with persistent spectral outliers

The previous synthetic example was constructed directly at the level of the spectrum of A. We now consider a more implementation-oriented test in which a cheap diagonal scaling is applied first but pronounced spectral outliers remain.

To this end, we construct an SPD matrix A so that its Jacobi-scaled form

$$B = D^{-1/2} A D^{-1/2}, \qquad D = \mathrm{diag}(A),$$

still contains both very small and comparatively large eigenvalues. More precisely, B is designed to have 30 eigenvalues near 10^{-6}, 15 eigenvalues near 10^{-10}, 4 comparatively

large eigenvalues in the range $[5, 20]$, and all remaining eigenvalues equal to 1. Thus, even after diagonal scaling, the spectrum is far from being well clustered.

The matrix size is $n = 2000$. For the low-rank methods, we use the same total rank budget, with $\ell_1 = 5$ and $\ell_2 = 50$, so that the Nyström preconditioner has rank $\ell_{\text{Nys}} = 55$.

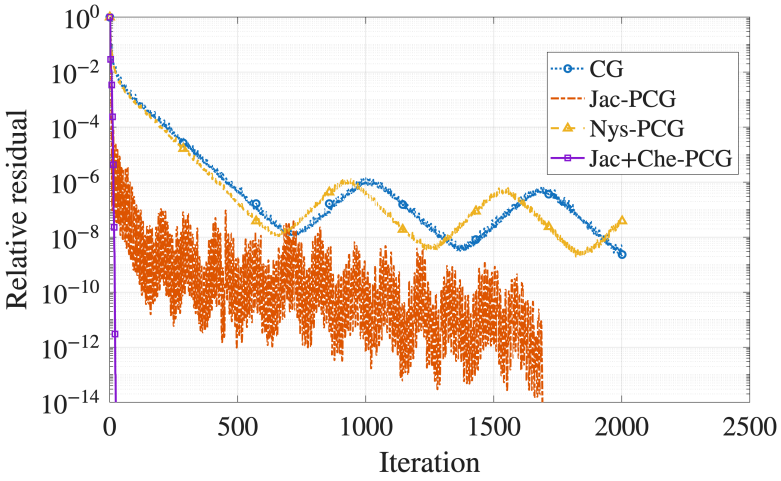


Fig. 7: Relative residual versus iteration for a synthetic matrix whose Jacobi-scaled form still contains pronounced lower-tail and upper-tail outliers.

Figure 7 shows that plain CG and Nys-PCG do not achieve satisfactory convergence within the iteration cap. Jac-PCG eventually converges, indicating that diagonal scaling is beneficial but insufficient for this example: it still requires 1689 iterations and 0.368 seconds in total. By contrast, Jac+Che-PCG reaches the prescribed tolerance in only 23 iterations, with 0.087 seconds for setup and 0.006 seconds for the PCG iteration, giving a total cost of 0.093 seconds.

This example highlights the main point of the two-stage strategy. Jacobi scaling provides a cheap first step, but it may not remove the extreme spectral components that dominate convergence. In the present test, the scaled operator still contains a pronounced lower tail together with a few upper outliers, so a low-rank approximation aimed mainly at the dominant subspace is not sufficient. The proposed method remains effective because its Chebyshev-filtered low-rank correction targets the remaining difficult modes, especially the near-null part of the spectrum.

6.3 Experiments on real-world data

We finally test the proposed preconditioner on matrices arising from a range of application areas. The test set is selected from the SuiteSparse Matrix Collection and includes symmetric positive definite matrices with challenging small-eigenvalue behavior. Their basic properties, including the dimension, number of nonzero entries, condition number, and application area, are listed in Table 1. For each matrix, we

compare CG, Jac-PCG, Nys-PCG, and the proposed Jac+Che-PCG under the same stopping criterion as in the synthetic experiments.

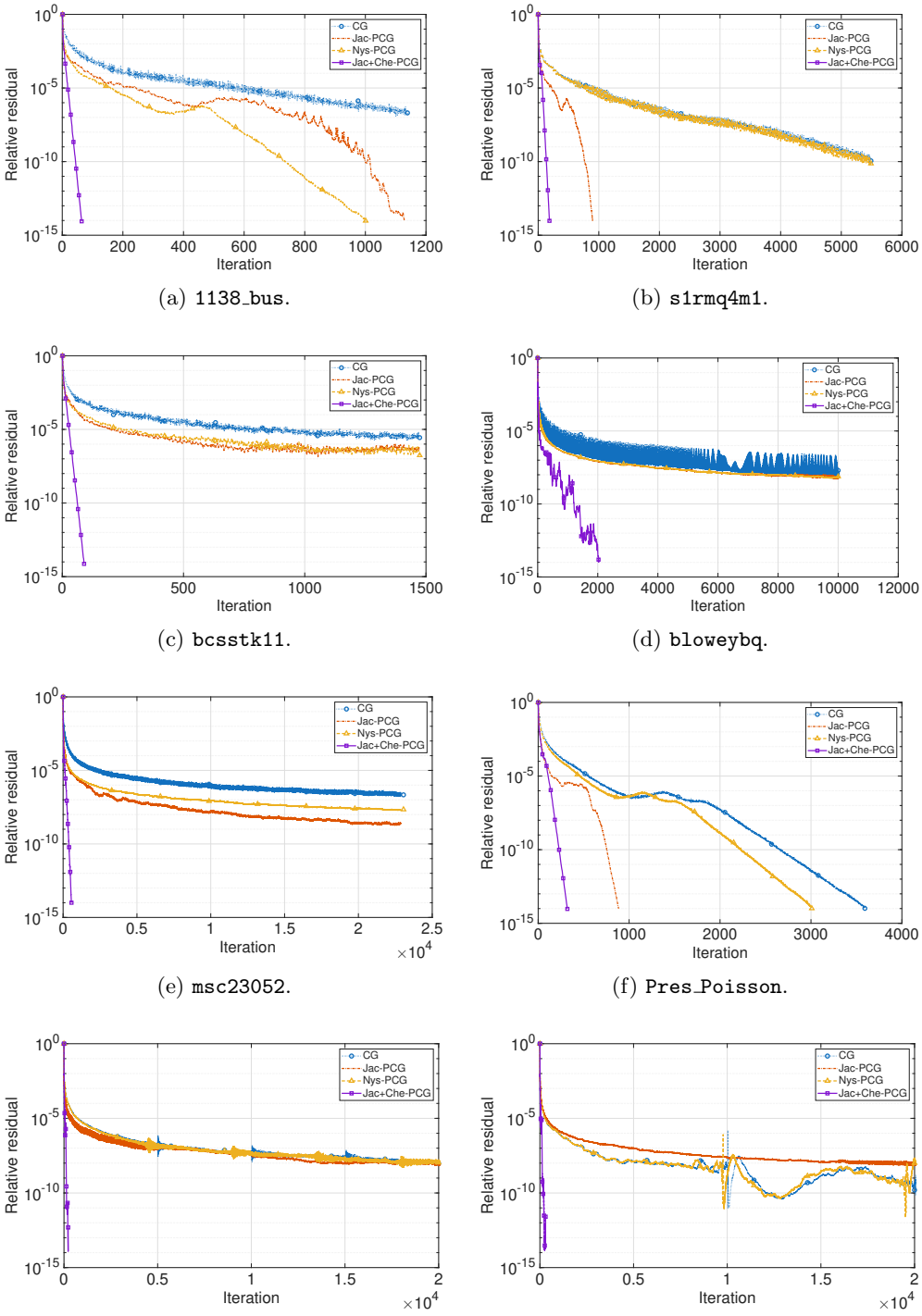
Table 1: Test matrices used in the real-data experiments.

Name	Size	nnz	$\kappa(A)$	Application field
1138_bus	1138	4054	8.572e+06	Power network
s1rmq4m1	5849	262411	1.810e+06	Structural
bcsstk11	1473	34241	2.212e+08	Structural
bloweybq	10001	49999	3.090e+18	Materials
msc23052	23052	1142686	7.433e+11	Structural
Pres_Poisson	14822	715804	2.037e+06	Computational Fluid Dynamics
LFAT5000	19994	79966	5.133e+17	Model reduction
LF10000	19998	99982	2.059e+18	Model reduction

Table 2 reports the numerical performance of the tested methods, including iteration counts, setup and solve times, and the spectral range of the preconditioned operator. The parameter column gives ℓ_{Nys} for Nys-PCG and the pair (ℓ_1, ℓ_2) for Jac+Che-PCG. All timings are reported in seconds. The setup time includes the construction of the corresponding low-rank preconditioner, the solve time refers to the subsequent PCG iteration, and the total time is the sum of the two.

The results in Table 2 show a consistent improvement at the lower end of the preconditioned spectrum. Jacobi scaling reduces the upper spectral scale for several matrices, but it often leaves very small eigenvalues near the origin. The Nyström preconditioner, which mainly captures a dominant subspace, shows a similar limitation on these examples. In contrast, the Jac+Che preconditioner substantially increases the smallest eigenvalue and produces much smaller condition numbers in the reported tests. For instance, on `1138_bus`, the condition number decreases from 4.903×10^5 with Jac-PCG to 20.941 with Jac+Che-PCG; on `bcsstk11`, it decreases to 73.318. Similar improvements are observed for the more ill-conditioned matrices `bloweybq`, `LFAT5000`, and `LF10000`. This behavior is consistent with the design of the method: the Chebyshev-filtered low-rank correction targets near-null spectral components that are not removed by diagonal scaling or by a dominant-subspace approximation.

The spectral improvement translates directly into iteration reduction. In all cases, Jac+Che-PCG converges within the prescribed iteration limit and requires substantially fewer iterations than the convergent baseline methods. On several ill-conditioned matrices, including `bcsstk11`, `bloweybq`, `msc23052`, `LFAT5000`, and `LF10000`, it is the only method among the four that reaches the prescribed tolerance within the iteration limit. Figure 8 displays the corresponding residual histories. The curves show that Jac+Che-PCG maintains a clear convergence advantage on the ill-conditioned test matrices.



(a) `1138_bus`. (b) `s1rmq4m1`.

(c) `bcsstk11`. (d) `bloweybq`.

(e) `msc23052`. (f) `Pres_Poisson`.

(g) `LFAT5000`. (h) `LF10000`.

Fig. 8: Relative residual curves versus iteration number for CG, Jac-PCG, Nys-PCG, and Jac+Che-PCG on eight application matrices.

Table 2: Spectral and runtime comparison on test matrices. Spectral quantities are computed from A for CG and from the corresponding preconditioned matrix for PCG methods. A dash "–" indicates nonconvergence. The parameter column reports ℓ_{Nys} for Nys-PCG and (ℓ_1, ℓ_2) for Jac+Che-PCG. Values $\lambda_{\min} \approx 0$ are treated as numerically singular, with condition number reported as ∞.

Matrix	Method	Param.	Spectrum			#Iter.	Time (s)		
			$\lambda_{\min}$	$\lambda_{\max}$	κ		Setup	Solve	Total
1138_bus	CG	–	3.517×10^{-3}	3.015×10^{4}	8.573×10^{6}	–	–	–	–
	Jac-PCG	–	4.079×10^{-6}	2.000	4.903×10^{5}	1127	0	0.023	0.023
	Nys-PCG	160	2.454×10^{-5}	7.132	2.907×10^{5}	1000	0.009	0.029	0.038
	Jac+Che-PCG	$(10, 150)$	**0.142**	2.970	**20.941**	**63**	0.033	**0.006**	0.039
s1rmq4m1	CG	–	0.380	6.874×10^{5}	1.811×10^{6}	–	–	–	–
	Jac-PCG	–	9.995×10^{-6}	4.315	4.317×10^{5}	898	0	0.081	0.081
	Nys-PCG	140	2.045×10^{-6}	3.492	1.708×10^{6}	–	–	–	–
	Jac+Che-PCG	$(20, 120)$	$\mathbf{9.516\times10^{-3}}$	12.885	$\mathbf{1.354\times10^{3}}$	**185**	0.129	**0.028**	0.157
bcsstk11	CG	–	2.964	6.556×10^{8}	2.212×10^{8}	–	–	–	–
	Jac-PCG	–	6.380×10^{-7}	3.769	5.907×10^{6}	–	–	–	–
	Nys-PCG	400	2.250×10^{-7}	3.579	1.591×10^{7}	–	–	–	–
	Jac+Che-PCG	$(10, 390)$	**0.104**	7.607	**73.318**	104	0.261	0.009	0.270
bloweybq	CG	–	1.618×10^{-15}	5.000×10^{3}	3.090×10^{18}	–	–	–	–
	Jac-PCG	–	3.563×10^{-16}	2.667	7.486×10^{15}	–	–	–	–
	Nys-PCG	2350	4.695×10^{-16}	2.916	6.210×10^{15}	–	–	–	–
	Jac+Che-PCG	$(50, 2300)$	$\mathbf{1.120\times10^{-7}}$	8.721	$\mathbf{7.789\times10^{7}}$	2025	9.140	5.210	14.349
msc23052	CG	–	9.554×10^{-4}	7.102×10^{8}	7.433×10^{11}	–	–	–	–
	Jac-PCG	–	6.226×10^{-9}	8.017	1.288×10^{9}	–	–	–	–
	Nys-PCG	4050	1.394×10^{-10}	5.306	3.808×10^{10}	–	–	–	–
	Jac+Che-PCG	$(50, 4000)$	$\mathbf{9.137\times10^{-4}}$	3.117	$\mathbf{3.411\times10^{3}}$	564	37.622	5.854	43.476
Pres_Poisson	CG	–	1.278×10^{-5}	26.029	2.037×10^{6}	3592	0	0.728	0.728
	Jac-PCG	–	2.939×10^{-5}	4.629	1.575×10^{5}	883	0	0.218	0.218
	Nys-PCG	110	7.542×10^{-6}	10.164	1.348×10^{6}	3008	0.050	1.139	1.189
	Jac+Che-PCG	$(10, 100)$	**0.011**	21.084	$\mathbf{1.849\times10^{3}}$	**320**	0.252	**0.135**	0.386
LFAT5000	CG	–	1.147×10^{-4}	5.888×10^{13}	5.133×10^{17}	–	–	–	–
	Jac-PCG	–	≈ 0	2.000	∞	–	–	–	–
	Nys-PCG	2850	1.551×10^{-17}	4.884	3.149×10^{17}	–	–	–	–
	Jac+Che-PCG	$(50, 2800)$	$\mathbf{4.599\times10^{-4}}$	6.810	$\mathbf{1.481\times10^{4}}$	220	15.890	1.387	17.277
LF10000	CG	–	2.288×10^{-4}	4.711×10^{14}	2.059×10^{18}	–	–	–	–
	Jac-PCG	–	≈ 0	2.000	∞	–	–	–	–
	Nys-PCG	3750	≈ 0	3.068	∞	–	–	–	–
	Jac+Che-PCG	$(50, 3700)$	$\mathbf{1.457\times10^{-4}}$	6.424	$\mathbf{4.407\times10^{4}}$	301	24.161	2.525	26.685

7 Conclusion

We have developed a Chebyshev-filtered randomized low-rank preconditioner for sparse SPD linear systems. The method is designed for problems in which a small number of extremal eigenvalues, especially near-null modes at the lower end of the spectrum, dominate the convergence of PCG. The practical construction combines an inexpensive base preconditioner with a low-rank correction for the extremal spectral components of the shaped operator. The upper tail is captured by a randomized dominant-subspace extraction, while the lower tail is treated by a Chebyshev-filtered

randomized sketch followed by a Rayleigh–Ritz compression. We establish subspace error bounds for the Chebyshev-filtered randomized extraction and relate the accuracy of the recovered extremal subspaces to the spectrum of the preconditioned operator. The resulting condition-number estimates show that, when the relevant spectral tails are captured with sufficient accuracy, the practical preconditioner yields a significantly improved preconditioned spectrum. Numerical experiments on synthetic and real-world matrices support these conclusions. Future work will consider adaptive selection of the ranks and problem-dependent choices of the lower-tail Chebyshev interval.

Funding ZD acknowledges financial support from the China Scholarship Council (CSC) under Project No. 202606260040. JY acknowledges support from the National Natural Science Foundation of China (Grant No. 12471357).

Data Availability The code to reproduce the numerical experiments may be found at https://github.com/dongzeyu67/chebyshev-randomized-preconditioner.

Declarations

Conflict of interest The authors have no conflict of interest to declare.

References

[1] Nicolaides, R.A.: Deflation of conjugate gradients with applications to boundary value problems. SIAM Journal on Numerical Analysis **24**(2), 355–365 (1987) https://doi.org/10.1137/0724027

[2] Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. Numerische Mathematik **126**(4), 741–770 (2014) https://doi.org/10.1007/s00211-013-0576-y

[3] Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM Journal on Numerical Analysis **20**(3), 626–637 (1983) https://doi.org/10.1137/0720042

[4] Hanke, M., Hansen, P.C.: Regularization methods for large-scale problems. Surveys on Mathematics for Industry **3**, 253–315 (1993)

[5] Powell, C.E., Elman, H.C.: Block-diagonal preconditioning for spectral stochastic finite-element systems. IMA Journal of Numerical Analysis **29**(2), 350–375 (2009) https://doi.org/10.1093/imanum/drn014

[6] Gittens, A., Mahoney, M.W.: Revisiting the Nyström method for improved large-scale machine learning. Journal of Machine Learning Research **17**(117), 1–65 (2016)

[7] Mahoney, M.W.: Randomized algorithms for matrices and data. Foundations and Trends in Machine Learning **3**(2), 123–224 (2011) https://doi.org/10.1561/2200000035

[8] Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards **49**(6), 409–436 (1952) https://doi.org/10.6028/jres.049.044

[9] Greenbaum, A.: Iterative Methods for Solving Linear Systems. SIAM, Philadelphia (1997). https://doi.org/10.1137/1.9781611970937

[10] Carpentieri, B., Duff, I.S., Giraud, L.: A class of spectral two-level preconditioners. SIAM Journal on Scientific Computing **25**(2), 749–765 (2003) https://doi.org/10.1137/s1064827502408591

[11] Tang, J.M., Nabben, R., Vuik, C., Erlangga, Y.A.: Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. Journal of Scientific Computing **39**(3), 340–370 (2009) https://doi.org/10.1007/s10915-009-9272-6

[12] Daas, H.A., Rees, T., Scott, J.: Two-level Nyström–Schur preconditioner for sparse symmetric positive definite matrices. SIAM Journal on Scientific Computing **43**(6), 3837–3861 (2021) https://doi.org/10.1137/21m139548x

[13] Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003). https://doi.org/10.1137/1.9780898718003

[14] Meijerink, J.A., Vorst, H.A.: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Mathematics of Computation **31**(137), 148–162 (1977) https://doi.org/10.2307/2005786

[15] Scott, J.A., Tůma, M.: HSL_MI28: An efficient and robust limited-memory incomplete Cholesky factorization code. ACM Transactions on Mathematical Software **40**(4), 24–12419 (2014) https://doi.org/10.1145/2617555

[16] Hook, J., Scott, J., Tisseur, F., Hogg, J.: A max-plus approach to incomplete Cholesky factorization preconditioners. SIAM Journal on Scientific Computing **40**(4), 1987–2004 (2018) https://doi.org/10.1137/16m1107735

[17] Grote, M.J., Huckle, T.: Parallel preconditioning with sparse approximate inverses. SIAM Journal on Scientific Computing **18**(3), 838–853 (1997) https://doi.org/10.1137/s1064827594276552

[18] Benzi, M., Tůma, M.: A sparse approximate inverse preconditioner for nonsymmetric linear systems. SIAM Journal on Scientific Computing **19**(3), 968–994 (1998) https://doi.org/10.1137/S1064827595294691

[19] Vuik, C., Segal, A., Meijerink, J.A.: An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. Journal of Computational Physics **152**(1), 385–403 (1999) https://doi.org/10.1006/jcph.1999.6255

[20] Vuik, C., Segal, A., Meijerink, J.A., Wijma, G.T.: The construction of projection vectors for a deflated ICCG method applied to problems with extreme contrasts in the coefficients. Journal of Computational Physics **172**(2), 426–450 (2001) https://doi.org/10.1006/jcph.2001.6795

[21] Dostál, Z.: Conjugate gradient method with preconditioning by projector. International Journal of Computer Mathematics **23**(3-4), 315–323 (1988) https://doi.org/10.1080/00207168808803625

[22] Mansfield, L.: On the use of deflation to improve the convergence of conjugate gradient iteration. Communications in Applied Numerical Methods **4**(1), 151–156 (1988) https://doi.org/10.1002/cnm.1630040202

[23] Mansfield, L.: Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers. SIAM Journal on Scientific and Statistical Computing **12**(6), 1314–1323 (1991) https://doi.org/10.1137/0912071

[24] Frank, J., Vuik, C.: On the construction of deflation-based preconditioners. SIAM Journal on Scientific Computing **23**(2), 442–462 (2001) https://doi.org/10.1137/s1064827500373231

[25] Nabben, R., Vuik, C.: A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners. Numerical Linear Algebra with Applications **15**(4-5), 355–372 (2008) https://doi.org/10.1002/nla.571

[26] Tang, J.M., MacLachlan, S.P., Nabben, R., Vuik, C.: A comparison of two-level preconditioners based on multigrid and deflation. SIAM Journal on Matrix Analysis and Applications **31**(4), 1715–1739 (2010) https://doi.org/10.1137/08072084x

[27] Gaul, A., Gutknecht, M.H., Liesen, J., Nabben, R.: A framework for deflated and augmented krylov subspace methods. SIAM Journal on Matrix Analysis and Applications **34**(2), 495–518 (2013) https://doi.org/10.1137/110820713

[28] Kahl, K., Rittich, H.: The deflated conjugate gradient method: Convergence, perturbation and accuracy. Linear Algebra and its Applications **515**, 111–129 (2017) https://doi.org/10.1016/j.laa.2016.10.027

[29] Xu, J.: Iterative methods by space decomposition and subspace correction. SIAM Review **34**(4), 581–613 (1992) https://doi.org/10.1137/1034116

[30] Toselli, A., Widlund, O.: Domain Decomposition Methods—Algorithms and Theory. Springer Series in Computational Mathematics, vol. 34. Springer, Berlin (2005). https://doi.org/10.1007/b137868

[31] Dolean, V., Jolivet, P., Nataf, F.: An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation. SIAM, Philadelphia (2015). https://doi.org/10.1137/1.9781611974065

[32] Spillane, N., Rixen, D.: Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. International Journal for Numerical Methods in Engineering **95**(11), 953–990 (2013) https://doi.org/10.1002/nme.4534

[33] Nataf, F., Xiang, H., Dolean, V., Spillane, N.: A coarse space construction based on local Dirichlet-to-Neumann maps. SIAM Journal on Scientific Computing **33**(4), 1623–1642 (2011) https://doi.org/10.1137/100796376

[34] Daas, H.A., Grigori, L.: A class of efficient locally constructed preconditioners based on coarse spaces. SIAM Journal on Matrix Analysis and Applications **40**(1), 66–91 (2019) https://doi.org/10.1137/18m1194365

[35] Daas, H.A., Grigori, L., Jolivet, P., Tournier, P.-H.: A multilevel Schwarz preconditioner based on a hierarchy of robust coarse spaces. SIAM Journal on Scientific Computing **43**(4), 1907–1928 (2021) https://doi.org/10.1137/19m1266964

[36] Li, R., Xi, Y., Saad, Y.: Schur complement-based domain decomposition preconditioners with low-rank corrections. Numerical Linear Algebra with Applications **23**(4), 706–729 (2016) https://doi.org/10.1002/nla.2051

[37] Xi, Y., Li, R., Saad, Y.: An algebraic multilevel preconditioner with low-rank corrections for sparse symmetric matrices. SIAM Journal on Matrix Analysis and Applications **37**(1), 235–259 (2016) https://doi.org/10.1137/15m1021830

[38] Zheng, Q., Xi, Y., Saad, Y.: A power Schur complement low-rank correction preconditioner for general sparse linear systems. SIAM Journal on Matrix Analysis and Applications **42**(2), 659–682 (2021) https://doi.org/10.1137/20m1316445

[39] Liberty, E., Woolfe, F., Martinsson, P.-G., Rokhlin, V., Tygert, M.: Randomized algorithms for the low-rank approximation of matrices. Proceedings of the National Academy of Sciences of the United States of America **104**(51), 20167–20172 (2007) https://doi.org/10.1073/pnas.0709640104

[40] Halko, N., Martinsson, P.-G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review **53**(2), 217–288 (2011) https://doi.org/10.1137/090771806

[41] Martinsson, P.-G., Tropp, J.A.: Randomized numerical linear algebra: Foundations and algorithms. Acta Numerica **29**, 403–572 (2020) https://doi.org/10.1017/s0962492920000021

[42] Saibaba, A.K.: Randomized subspace iteration: Analysis of canonical angles and unitarily invariant norms. SIAM Journal on Matrix Analysis and Applications **40**(1), 23–48 (2019) https://doi.org/10.1137/18m1179432

[43] Nakatsukasa, Y.: Fast and stable randomized low-rank matrix approximation. arXiv preprint arXiv:2009.11392 (2020)

[44] Nyström, E.J.: Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. Acta Mathematica **54**, 185–204 (1930) https://doi.org/10.1007/bf02547521

[45] Frangella, Z., Tropp, J.A., Udell, M.: Randomized Nyström preconditioning. SIAM Journal on Matrix Analysis and Applications **44**(2), 718–752 (2023) https://doi.org/10.1137/21m1466244

[46] Hageman, L.A., Young, D.M.: Applied Iterative Methods. Academic Press, New York (1981)

[47] Golub, G.H., Kent, M.D.: Estimates of eigenvalues for iterative methods. Mathematics of Computation **53**(188), 619–626 (1989) https://doi.org/10.2307/2008724

[48] Fischer, B.: Polynomial Based Iteration Methods for Symmetric Linear Systems. Wiley–Teubner, Stuttgart (1996). https://doi.org/10.1007/978-3-663-11108-5

[49] Sorensen, D.C.: Implicit application of polynomial filters in a k-step Arnoldi method. SIAM Journal on Matrix Analysis and Applications **13**(1), 357–385 (1992) https://doi.org/10.1137/0613025

[50] Arioli, M., Ruiz, D.: A Chebyshev-based two-stage iterative method as an alternative to the direct solution of linear systems. Technical Report RAL-TR-2002-021, Rutherford Appleton Laboratory (2002)

[51] Arioli, M., Ruiz, D.: Flexible deflation in Krylov methods with Chebyshev-based polynomial filters. Technical Report RAL-TR-2009-014, Rutherford Appleton Laboratory (2009)

[52] Giraud, L., Ruiz, D., Touhami, A.: Krylov and polynomial iterative solvers combined with partial spectral factorization for SPD linear systems. In: High Performance Computing for Computational Science – VECPAR 2004. Lecture Notes in Computer Science, vol. 3402, pp. 637–656. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/11403937_48

[53] Giraud, L., Ruiz, D., Touhami, A.: A comparative study of iterative solvers exploiting spectral information for SPD systems. SIAM Journal on Scientific Computing **27**(5), 1760–1786 (2006) https://doi.org/10.1137/040608301

[54] Golub, G.H., Ruiz, D., Touhami, A.: A hybrid approach combining Chebyshev filter and conjugate gradient for solving linear systems with multiple right-hand sides. SIAM Journal on Matrix Analysis and Applications **29**(3), 774–795 (2007) https://doi.org/10.1137/060649458

[55] Boucheron, S., Lugosi, G., Massart, P.: Concentration Inequalities: A Nonasymptotic Theory of Independence. Oxford University Press, Oxford (2013). https://doi.org/10.1093/acprof:oso/9780199535255.001.0001

[56] Vershynin, R.: High-Dimensional Probability: An Introduction with Applications in Data Science. Cambridge University Press, Cambridge (2018). https://doi.org/10.1017/9781108231596

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

51/2026 Bellezza P.; Ciaramella G.; Macchini C.; Mazzieri I.; Verani M.
ParaFlow: Parareal Acceleration of Gradient-Flow Minimization

52/2026 Bonazzoli M.; Ciaramella G.; Mazzieri I.
On the Unmapped Tent Pitching for the Heterogeneous Wave Equation

50/2026 Donnarumma, A.; Guagliardi, O.; Di Stazio F.; Mazza E.; Tanelli M.; Paganoni A.M.
Modelling Well-Being and Psychological Risk in Doctoral Education: An Integrated Latent Trait Approach

49/2026 Bortolotti, T.; Troilo, R.; Casu, F.; Vantini, S.; Menafoglio, A.
Regularized covariance estimation from partially observed interferometric data

48/2026 Antonietti, P.F.; Corti, M.; Orlando, G.
Optimized high-order IMEX-RK schemes for degenerate diffusion-reaction problems with application to travelling waves phenomena

47/2026 Torri, V.; Barbieri, E.; Cantarutti, A.; Giaquinto, C.; Ieva, F.
Automatic identification of diagnosis from hospital discharge letters via weakly supervised Natural Language Processing

46/2026 Cancrini, A.; Ciaramella, G.; Antonietti, P.F.
A Scalable Deflated Conjugate Gradient Solver for the Time-Dependent Pseudo-Stress Stokes Problem

45/2026 Antonietti, P.F.; Botti, M.; Parolini, N.; Pederzoli, V.; Verani, M.
Polytopal Discontinuous Galerkin Discretizations of Coupled Non-Newtonian Stokes-Darcy Systems

44/2026 Bonetti, S; Botti, M.; Antonietti, P.F.
Splitting strategies for the fully-coupled nonlinear thermo-hydro-mechanical problem

43/2026 Micheletti, S.
A validated MATLAB framework for sparse vectorized finite element assembly