Electronic Transactions on Numerical Analysis.
Volume 65, pp. 63–92, 2026.
© 2026, the Author(s), licensed via CC BY 4.0.
ISSN 1068–9613.
DOI: 10.1553/etna_vol65s63

ETNA
Kent State University and
Johann Radon Institute (RICAM)

PRECONDITIONING WITHOUT A PRECONDITIONER USING RANDOMIZED BLOCK KRYLOV SUBSPACE METHODS[*]

TYLER CHEN[†], CAROLINE HUBER[‡], ETHAN LIN[‡], AND HAJAR ZAID[§]

Abstract. We describe a randomized variant of the block conjugate gradient method for solving a single positive definite linear system of equations. This method provably outperforms the preconditioned conjugate gradient method with a broad class of Nyström-based preconditioners, without ever explicitly constructing a preconditioner. In analyzing our algorithm, we derive theoretical guarantees for new variants of the Nyström-preconditioned conjugate gradient method, which may be of separate interest. We also describe how our approach yields fast algorithms for key data-science tasks such as computing the entire ridge regression regularization path and generating multiple independent samples from a high-dimensional Gaussian distribution.

Key words. preconditioning, randomized, conjugate gradient, block Krylov subspace methods

AMS subject classifications. 65F08, 65F60, 65F50, 68Q25

1. Introduction.

Solving the regularized linear system

$$\mathbf{A}_\mu \mathbf{x} = \mathbf{b}, \qquad \mathbf{A}_\mu := \mathbf{A} + \mu \mathbf{I}, \tag{1.1}$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$ is symmetric positive definite and $\mu \geq 0$, is a critical task across the computational sciences. Systems of the form (1.1) arise in a variety of settings, including the following:

Positive definite linear systems. When $\mu = 0$, (1.1) is simply the task of solving a positive definite linear system $\mathbf{Ax} = \mathbf{b}$, one of the most common problems in numerical linear algebra.

Ridge regression. Given a data matrix $\mathbf{Z} \in \mathbb{R}^{n\times d}$, a source term $\mathbf{f} \in \mathbb{R}^n$, and a regularization parameter $\mu \geq 0$, the ridge regression problem[1] is to find a minimizer $\mathbf{x}$ of

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left(\|\mathbf{Zx} - \mathbf{f}\|^2 + \mu \|\mathbf{x}\|^2 \right). \tag{1.2}$$

A direct computation shows that the solution to (1.2) is also the solution to

$$(\mathbf{Z}^\mathsf{T}\mathbf{Z} + \mu\mathbf{I})\mathbf{x} = \mathbf{Z}^\mathsf{T}\mathbf{f},$$

i.e., a system of the form (1.1) with $\mathbf{A} = \mathbf{Z}^\mathsf{T}\mathbf{Z}$ and $\mathbf{b} = \mathbf{Z}^\mathsf{T}\mathbf{f}$. In a number of applications, we are interested in the whole regularization path, i.e., $\mathbf{x} = \mathbf{x}_\mu$, for all values $\mu \geq 0$, for instance, in order to perform cross-validation to select a value μ for future predictions.

Sampling Gaussians. If $\mathbf{b}$ has independent standard normal entries, then $\boldsymbol{\mu} + \mathbf{A}^{1/2}\mathbf{b}$ is a Gaussian vector with mean $\boldsymbol{\mu}$ and covariance $\mathbf{A}$. In order to approximate $\mathbf{A}^{1/2}\mathbf{b}$, it is common to make use of the identity

$$\mathbf{A}^{1/2}\mathbf{b} = \frac{2}{\pi}\int_0^\infty \mathbf{A}(\mathbf{A} + z^2\mathbf{I})^{-1}\mathbf{b}\,\mathrm{d}z. \tag{1.3}$$

The factor $(\mathbf{A} + z^2\mathbf{I})^{-1}\mathbf{b}$ in the integrand of (1.3) is of the form (1.1) with $\mu = z^2$. Often we are interested in sampling several Gaussian vectors with the same covariance matrix.

[*]Received August 25, 2025. Accepted December 6, 2025. Published online on February 3, 2026. Recommended by Lothar Reichel.

[†]Corresponding author. JPMorganChase (work on this paper was initiated while at New York University) (tylerchen.research@gmail.com).

[‡]New York University, New York.

[§]Graduate Center, CUNY, New York.

[1]Ridge regression is a special case of Tikhonov regularization, where $\mu\|\mathbf{x}\|^2$ is replaced by a somewhat more general regularization term $\|\boldsymbol{\Gamma}\mathbf{x}\|^2$ [22].

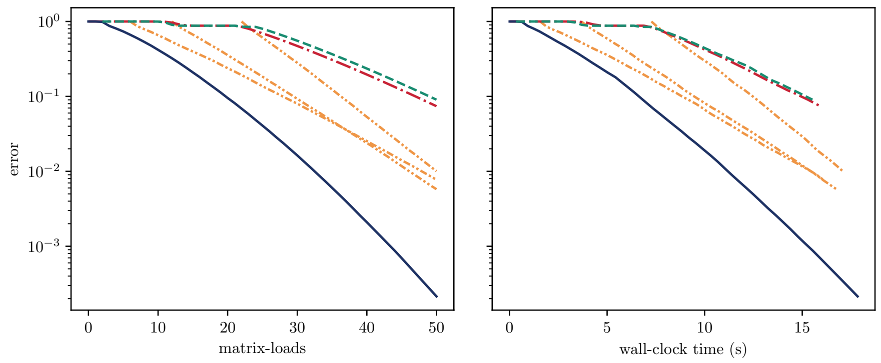


FIG. 1.1. *The relative error* $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}}/\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ *in terms of the matrix loads (left) and the wall-clock time (right) for our proposed randomized variant of the block conjugate gradient method (——), the standard conjugate gradient method (–·–), the Nyström-preconditioned conjugate gradient method from [25] (– – –), and the generalization of the Nyström-preconditioned conjugate gradient method using higher-depth Nyström approximations (–··–). Our method outperforms all these methods without the need for selecting hyperparameters (see Theorem 3.3), which may be difficult to do effectively in practice. In particular, we store* $\mathbf{A}$ *in 8 separate* 1000×8000 *chunks and perform (block) matrix-vector products with* $\mathbf{A}$ *by sequentially loading a single chunk from the disk into random-access memory and performing the appropriate part of the products. The runtime is dominated by the cost of loading chunks of the matrix into memory, so the wall-clock-time is nearly proportional to the matrix loads. A full description of the experiment is given in Section 7.1.*

1.1. (Block) Krylov subspace methods.

All three problems described above are commonly solved using a class of algorithms called Krylov subspace methods (KSMs); see, e.g., [8, 39, 63]. KSMs make use of the so-called Krylov subspace

$$\mathcal{K}_t(\mathbf{A}, \mathbf{b}) := \operatorname{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{t-1}\mathbf{b}\},$$

which can be efficiently computed using matrix-vector products with $\mathbf{A}$. Since it is often of interest to solve (1.1) for multiple values of μ, we repeatedly make use of the fact that the Krylov subspace is shift-invariant, i.e., that

$$\forall \mu \in \mathbb{C} : \mathcal{K}_t(\mathbf{A}_\mu, \mathbf{b}) = \mathcal{K}_t(\mathbf{A}, \mathbf{b}). \tag{1.4}$$

In this paper, we advocate for the use of generalizations of KSMs, called block-KSMs. Given a matrix $\mathbf{B} \in \mathbb{R}^{d \times m}$ (typically $m \ll d$) with columns $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m)}$, the block Krylov subspace is defined as

$$\mathcal{K}_t(\mathbf{A}, \mathbf{B}) := \mathcal{K}_t(\mathbf{A}, \mathbf{b}^{(1)}) + \cdots + \mathcal{K}_t(\mathbf{A}, \mathbf{b}^{(m)}). \tag{1.5}$$

That is, $\mathcal{K}_t(\mathbf{A}, \mathbf{B})$ is the space consisting of all linear combinations of vectors in the Krylov subspaces $\mathcal{K}_t(\mathbf{A}, \mathbf{b}^{(1)}), \ldots, \mathcal{K}_t(\mathbf{A}, \mathbf{b}^{(m)})$. Analogous to (1.4), the block Krylov subspace is shift-invariant, i.e., $\mathcal{K}_t(\mathbf{A}_\mu, \mathbf{B}) = \mathcal{K}_t(\mathbf{A}, \mathbf{B})$ for any shift μ.

As we discuss in Section 1.2, compared to their single-vector counterparts, block-KSMs have a number of computational benefits on modern computational architectures. As such, block-KSMs are widely used for tasks in numerical linear algebra, such as low-rank approximation and eigenvalue approximation, for which they are known to satisfy strong theoretical guarantees [31, 38, 43, 46, 51, 62, 68].

Interestingly, the use of block-KSMs for solving problems like (1.1) is more limited (see Section 6 for a discussion), and single-vector methods such as the conjugate gradient algorithm are more popular. A high-level aim of this paper is to encourage further study on the use of block-KSMs for tasks such as (1.1). In particular, we believe that techniques from randomized numerical linear algebra have the potential to provide new insights and theoretical guarantees for block-KSMs used to solve (1.1).

1.2. Computational assumptions. Throughout this paper, we assume $\mathbf{A}$ is accessed only through matrix-vector products $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ or block matrix-vector products $[\mathbf{x}_1 \cdots \mathbf{x}_m] \mapsto [\mathbf{A}\mathbf{x}_1 \cdots \mathbf{A}\mathbf{x}_m]$. While one can simulate a block matrix-vector product using m matrix-vector products, in many settings the cost of block products is often nearly independent of the block size m (as long as m is not too large). For example, this can be the case when $\mathbf{A}$ is so large that it must be loaded in chunks from slow memory (as in Figure 1.1), where $\mathbf{A}$ is the Hessian of a large neural network and matrix-vector products are performed using "Pearlmutter's trick" [58], or where $\mathbf{A}$ corresponds to the solution operator of an integral equation and is applied via a fast direct solver [45]. Allowing for more efficient data access is widely understood as one of the major benefits of many randomized linear algebra algorithms [46, 68].

The algorithms we propose in this paper are designed to economize the number of times that $\mathbf{A}$ is loaded into memory (henceforth referred to as matrix loads) and not with respect to other costs such as matrix-vector products, floating-point operations, or storage. In settings where matrix loads are not the dominant cost, the proposed algorithms may not yield significant advantages. The costs of the algorithms from this paper are further discussed in Section 3.2, and we explore them in the numerical experiments in Section 7. A detailed understanding of tradeoffs for various costs is somewhat beyond the scope of this paper but is of practical importance.

1.3. Notation. We denote the eigenvalues of $\mathbf{A}$ by $\lambda_1 \geq \cdots \geq \lambda_d$. We write $\kappa = \lambda_1/\lambda_d$, and let $\kappa_{r+1}(\mu) := (\lambda_{r+1} + \mu)/(\lambda_d + \mu)$ be the condition number of $\mathbf{A}_\mu$ with the top-r eigenvalues removed. The eigenvalues of an arbitrary positive definite matrix $\mathbf{M}$ are $\lambda_1(\mathbf{M}) \geq \cdots \geq \lambda_d(\mathbf{M})$. The condition number of $\mathbf{M}$ is $\kappa(\mathbf{M}) := \lambda_1(\mathbf{M})/\lambda_d(\mathbf{M})$, and the spectral norm is $\|\mathbf{M}\| := \lambda_1(\mathbf{B})$. The $\mathbf{M}$-norm of a vector $\mathbf{x}$ is defined as $\|\mathbf{x}\|_{\mathbf{M}} := \|\mathbf{M}^{1/2}\mathbf{x}\| = \sqrt{\mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}}$.

1.4. Organization. In Section 2 we describe key background on Krylov subspace methods and preconditioning. Our algorithm and main conceptual results are described in Section 3. Explicit and simple-to-use probabilistic results are proved in Section 4. We discuss our bounds in the context of previous work in Section 6 and show how the theory we develop can be used to sample Gaussian vectors in Section 5. Finally, numerical experiments are provided in Section 7.

2. Background.

In this section, we provide an overview of a number of relevant algorithms. The methods we discuss in this section are standard, and for mathematical simplicity, we define them by their optimality conditions rather than algorithmically (all of our bounds assume exact arithmetic); see, e.g., [33, 48, 63] for comprehensive treatments of these methods.

2.1. The preconditioned conjugate gradient method. The preconditioned conjugate gradient (PCG) algorithm is a powerful KSM for solving positive definite linear systems. It makes use of a symmetric positive definite preconditioner $\mathbf{P}_\mu$ that transforms the system $\mathbf{A}_\mu\mathbf{x} = \mathbf{b}$ into the system

$$(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})\mathbf{y} = \mathbf{P}_\mu^{-1/2}\mathbf{b}, \qquad \mathbf{P}_\mu^{1/2}\mathbf{x} = \mathbf{y},$$

and operates over a transformed Krylov subspace

$$\mathcal{K}_t(\mathbf{A}_\mu, \mathbf{b}; \mathbf{P}_\mu) := \mathcal{K}_t(\mathbf{P}_\mu^{-1}\mathbf{A}_\mu, \mathbf{P}_\mu^{-1}\mathbf{b}). \tag{2.1}$$

Specifically, the PCG method is defined by an optimality condition.

DEFINITION 2.1. *The t-th PCG iterate corresponding to a positive definite preconditioner $\mathbf{P}_\mu$ is defined as*

$$\mathsf{pcg}_t(\mu) := \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}_t(\mathbf{A}_\mu,\mathbf{b};\mathbf{P}_\mu)} \|\mathbf{A}_\mu^{-1}\mathbf{b}-\mathbf{x}\|_{\mathbf{A}_\mu}.$$

The iterate $\mathsf{pcg}_t(\mu)$ can be computed using $t-1$ matrix-vector products with $\mathbf{A}$ (and t products with $\mathbf{P}_\mu^{-1}$).

The simplest choice of a preconditioner is $\mathbf{P}_\mu = \mathbf{I}$, which yields the standard conjugate gradient (CG) algorithm [40].

DEFINITION 2.2. *The t-th CG iterate is defined as*

$$\mathsf{cg}_t(\mu) := \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}_t(\mathbf{A}_\mu,\mathbf{b})} \|\mathbf{A}_\mu^{-1}\mathbf{b}-\mathbf{x}\|_{\mathbf{A}_\mu}.$$

2.1.1. Convergence. The PCG (and hence the CG) algorithm satisfies a well-known convergence guarantee in terms of the condition number of $\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}$; see, e.g., [33].[2]

COROLLARY 2.3. *Let $\mathbf{P}_\mu$ be any preconditioner. Then the t-th preconditioned-CG iterate corresponding to the preconditioner $\mathbf{P}_\mu$ satisfies*

$$\frac{\|\mathbf{A}_\mu^{-1}\mathbf{b}-\mathsf{pcg}_t(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \leq 2\exp\left(-\frac{2t}{\sqrt{\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})}}\right).$$

Corollary 2.3 implies that if we can find $\mathbf{P}_\mu$ such that $\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})$ is small, then the PCG iteration converges rapidly. The choice of $\mathbf{P}_\mu$ minimizing this condition number is, of course, $\mathbf{P}_\mu = \mathbf{A}_\mu$, but this is not a practical preconditioner; if we knew $\mathbf{A}_\mu^{-1}$, then we could easily compute the solution $\mathbf{A}_\mu^{-1}\mathbf{b}$. Thus, finding a suitable choice of $\mathbf{P}_\mu$ that balances improvements in the convergence of the PCG method with the cost of building/applying $\mathbf{P}_\mu$ is critical.

2.2. Deflation preconditioners. If $\mathbf{A}$ is poorly conditioned due to the presence of r eigenvalues much larger than the remaining $n-r$ eigenvalues, then we might hope to learn a good approximation of the top-r eigenvalues and "correct" this ill-conditioning. Towards this end, suppose we have a good rank-r approximation $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ of $\mathbf{A}$, where $\mathbf{U}$ has orthonormal columns and $\mathbf{D}$ is diagonal. Intuitively, this low-rank approximation contains the information needed to remove the top eigenvalues of $\mathbf{A}$, thereby reducing the condition number of the preconditioned system. In particular, one can form the preconditioner

$$\mathbf{P}_\mu := \frac{1}{\theta+\mu}\mathbf{U}(\mathbf{D}+\mu\mathbf{I})\mathbf{U}^\mathsf{T} + (\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T}), \tag{2.2}$$

where $\theta > 0$ is a parameter that must be chosen along with the factorization $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$.

It is not hard to verify that

$$\mathbf{P}_\mu^{-1} = (\theta+\mu)\mathbf{U}(\mathbf{D}+\mu\mathbf{I})^{-1}\mathbf{U}^\mathsf{T} + (\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T}).$$

This means that applying $\mathbf{P}_\mu^{-1}$ to a vector can be done in $O(dr)$ arithmetic operations, which is relatively cheap if $r \ll d$. Hence, as long as a reasonable factorization $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ can be obtained, (2.2) can be used as a preconditioner; see [26, 37] and the references therein.

[2] Here we use that $((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1))^t \leq \exp(-2t/\sqrt{\kappa})$ for all $\kappa \geq 1$ in order to obtain a simpler expression than the typical bound. This approximation is pessimistic when $\kappa \approx 1$, but it is clear from our analysis that other bounds for the CG method can be used instead.

2.2.1. Exact deflation. Traditionally, it has been suggested to take $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ as $[\![\mathbf{A}]\!]_r$, i.e., the *best* rank-r approximation of $\mathbf{A}$, that is, to set $\mathbf{D}$ as the diagonal matrix containing the top-r eigenvalues of $\mathbf{A}$ and $\mathbf{U}$ containing the corresponding eigenvectors. In this case, (2.2) is sometimes called a spectral deflation preconditioner. The name arises because the eigenvalues of $\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}$ are

$$\underbrace{\theta+\mu,\dots,\theta+\mu}_{r\text{-copies}},\lambda_{r+1}+\mu,\dots,\lambda_d+\mu,$$

and hence, if $\theta \in [\lambda_d, \lambda_{r+1}]$, then the condition number of $\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}$ is $\kappa_{r+1}(\mu) = (\lambda_{r+1}+\mu)/(\lambda_d+\mu)$. In other words, the top-r eigenvalues of $\mathbf{A}_\mu$ are deflated. In combination with Corollary 2.3, this yields the following convergence guarantee:

COROLLARY 2.4. *Let $\mathbf{P}_\mu$ be the preconditioner (2.2) corresponding to $[\![\mathbf{A}]\!]_r$, the rank-r truncated SVD of $\mathbf{A}$, and let $\theta \in [\lambda_d, \lambda_{r+1}]$. Then the t-th PCG iterate (Definition 2.1) corresponding to the preconditioner $\mathbf{P}_\mu$ satisfies*

$$\frac{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{pcg}_t(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \le 2\exp\left(-\frac{2t}{\sqrt{\kappa_{r+1}(\mu)}}\right).$$

When $\lambda_{r+1} \ll \lambda_1$ and μ is not too large, then the rate of convergence guaranteed by the bound can be much faster than without preconditioning ($r = 0$).[3] Of course, any potential benefits with respect to convergence must be weighted against the cost of constructing the spectral deflation preconditioner, and exact deflation, which requires computing the top eigenvectors of $\mathbf{A}$, can be costly.

2.2.2. Nyström preconditioning. Techniques from randomized numerical linear algebra allow near-optimal low-rank approximations of $\mathbf{A}$ to be computed very efficiently [38, 68], so we might hope that the corresponding preconditioner works nearly as well as spectral deflation while being much cheaper to compute.

In particular, it is reasonable to take $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ as the eigendecomposition of the randomized block-Krylov Nyström approximation, *mathematically* defined as

$$\mathbf{A}\langle\mathbf{K}_s\rangle := (\mathbf{A}\mathbf{K}_s)(\mathbf{K}_s^\mathsf{T}\mathbf{A}\mathbf{K}_s)^\dagger(\mathbf{K}_s^\mathsf{T}\mathbf{A}), \tag{2.3}$$

where $\mathbf{K}_s := [\boldsymbol{\Omega}\ \mathbf{A}\boldsymbol{\Omega}\ \cdots\ \mathbf{A}^{s-1}\boldsymbol{\Omega}] \in \mathbb{R}^{d\times(s\ell)}$ and $\boldsymbol{\Omega} \in \mathbb{R}^{d\times\ell}$ is a matrix of independent standard normal random variables.[4] This variant of the Nyström approximation is among the most powerful randomized low-rank approximation algorithms and can be implemented using s matrix loads. Note, however, that the algorithm should not be implemented as written in (2.3). In particular, one should avoid using a monomial basis $\mathbf{K}_s$ and carefully structure the interactions with $\mathbf{A}$ to avoid unnecessary costs; see, e.g., [68] for a pseudocode.

Nyström-based preconditioning is effective in theory and practice and has been an active area of research in recent years [6, 16, 21, 25, 41, 46, 74]. Most related to this paper is the theoretical analysis of [25], which proves that if $s = 1$, $\theta = \lambda_\ell(\mathbf{A}\langle\boldsymbol{\Omega}\rangle)$, and the sketching dimension ℓ is of the order of the effective dimension $d_{\text{eff}}(\mu) := \operatorname{tr}(\mathbf{A}\mathbf{A}_\mu^{-1}) = \sum_{i=1}^d \lambda_i/(\lambda_i+\mu)$, then the Nyström PCG method converges in at most $t = O(\log(1/\varepsilon))$ iterations, i.e., independent of any spectral properties of $\mathbf{A}$. Our analysis makes use of the same general techniques as

[3] The standard CG method also satisfies bounds in terms of λ_{r+1}/λ_d, at least in exact arithmetic; see Theorem B.1 and the discussion in Appendix B.

[4] In this paper, we only consider Nyström preconditioning where $\boldsymbol{\Omega}$ is a Gaussian matrix. Variants of a Nyström approximation based on subsampling rows/columns of $\mathbf{A}$ are effective in some settings (e.g., kernel ridge regression); see [25, §2.2.3] for a discussion.

those in [25] but is applicable when $s > 1$ as well as when $\mu = 0$. We discuss the theoretical bounds of [25] in Appendix B.2.

3. Our approach: the augmented block-CG method.

In this paper we advocate for the direct application of block KSMs to (1.1). The concept of block Krylov subspaces (1.5) naturally gives rise to the block-CG algorithm [54].

DEFINITION 3.1. *Let* $\mathbf{B} = [\mathbf{b}^{(1)} \cdots \mathbf{b}^{(m)}]$. *The* t*-th block-CG iterates are defined as*

$$\mathsf{bcg}_t^{(i)}(\mu) := \operatorname*{argmin}_{\mathbf{x} \in \mathcal{K}_t(\mathbf{A}_\mu, \mathbf{B})} \|\mathbf{A}_\mu^{-1}\mathbf{b}^{(i)} - \mathbf{x}\|_{\mathbf{A}_\mu}.$$

The block-CG iterates $\mathsf{bcg}_t^{(1)}(\mu), \ldots, \mathsf{bcg}_t^{(m)}(\mu)$ can be simultaneously computed using $t-1$ block matrix-vector products with $\mathbf{A}$. We further discuss the implementation and the costs in Section 3.2.

3.1. Implicit preconditioning.

Our main conceptual result is that by augmenting $\mathbf{b}$ with $\mathbf{\Omega}$, the block-CG method implicitly enjoys the benefits of certain classes of preconditioners built using $\mathbf{\Omega}$. We begin with a key observation about the relation between the block Krylov subspace (1.5) and the preconditioned Krylov subspace (2.1) corresponding to a certain class of preconditioners.

THEOREM 3.2. *Suppose that* $\mathbf{P}_\mu = (\mathbf{I}+\mathbf{X})^{-1}$, *where* $\operatorname{range}(\mathbf{X}) \subseteq \mathcal{K}_{s+1}(\mathbf{A}_\mu, \mathbf{\Omega})$. *Then,*

$$\mathcal{K}_t(\mathbf{A}_\mu, \mathbf{b}; \mathbf{P}_\mu) \subseteq \mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \mathbf{\Omega}).$$

The basic idea is simple. By definition, $\mathcal{K}_t(\mathbf{A}_\mu, \mathbf{b}; \mathbf{P}_\mu)$ consists of linear combinations of the vectors

$$(\mathbf{P}_\mu^{-1}\mathbf{A}_\mu)^k \mathbf{P}_\mu^{-1}\mathbf{b} = ((\mathbf{I}+\mathbf{X})\mathbf{A}_\mu)^k(\mathbf{I}+\mathbf{X})\mathbf{b}, \qquad k = 0, 1, \ldots, t-1,$$

and each $((\mathbf{I}+\mathbf{X})\mathbf{A}_\mu)^k(\mathbf{I}+\mathbf{X})\mathbf{b}$ can be expressed as a linear combination of vectors that live in the specified space.

Proof. By the shift invariance of (block) Krylov subspaces, we have

$$\mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \Omega) = \mathcal{K}_t(\mathbf{A}_\mu, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}_\mu, \mathbf{\Omega}).$$

Hence, without loss of generality it suffices to consider the case $\mu = 0$. For notational simplicity, we denote $\mathbf{P}_\mu$ by $\mathbf{P}$.

We proceed by induction, beginning with the base case $t = 1$. Observe that,

$$\mathcal{K}_1(\mathbf{P}^{-1}\mathbf{A}, \mathbf{P}^{-1}\mathbf{b}) = \operatorname{span}(\mathbf{P}^{-1}\mathbf{b}) = \operatorname{span}((\mathbf{I}+\mathbf{X})\mathbf{b}) = \operatorname{span}(\mathbf{b}+\mathbf{X}\mathbf{b}).$$

Clearly $\mathbf{b} \in \mathcal{K}_1(\mathbf{A}, \mathbf{b})$ and $\mathbf{X}\mathbf{b} \in \operatorname{range}(\mathbf{X}) \subseteq \mathcal{K}_{s+1}(\mathbf{A}, \mathbf{\Omega})$, so it follows that $\mathbf{b}+\mathbf{X}\mathbf{b} \subseteq \mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \mathbf{\Omega})$ as desired.

Now, assume that

$$\mathcal{K}_{t-1}(\mathbf{P}^{-1}\mathbf{A}, \mathbf{P}^{-1}\mathbf{b}) \subseteq \mathcal{K}_{t-1}(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{(t-1)+s}(\mathbf{A}, \mathbf{\Omega}).$$

We consider the order-t subspace

$$\begin{aligned}\mathcal{K}_t(\mathbf{P}^{-1}\mathbf{A}, \mathbf{P}^{-1}\mathbf{b}) = \operatorname{span}\{&\mathbf{P}^{-1}\mathbf{b}, (\mathbf{P}^{-1}\mathbf{A})\mathbf{P}^{-1}\mathbf{b}, \ldots, \\ &(\mathbf{P}^{-1}\mathbf{A})^{t-2}\mathbf{P}^{-1}\mathbf{b}, (\mathbf{P}^{-1}\mathbf{A})^{t-1}\mathbf{P}^{-1}\mathbf{b}\}.\end{aligned}$$

From the inductive hypothesis, we know that for all $j = 0, ..., t-2$,

$$\begin{aligned}(\mathbf{P}^{-1}\mathbf{A})^j\mathbf{P}^{-1}\mathbf{b} \in \mathcal{K}_{t-1}(\mathbf{P}^{-1}\mathbf{A}, \mathbf{P}^{-1}\mathbf{b}) &\subseteq \mathcal{K}_{t-1}(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s-1}(\mathbf{A}, \mathbf{\Omega}) \\ &\subseteq \mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \mathbf{\Omega}),\end{aligned}$$

where the last inclusion follows from the nested property of Krylov subspaces.

Thus, it remains to show that $(\mathbf{P}^{-1}\mathbf{A})^{t-1}\mathbf{P}^{-1}\mathbf{b} \in \mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \mathbf{\Omega})$. To this end, let $\mathbf{v} = (\mathbf{P}^{-1}\mathbf{A})^{t-2}\mathbf{P}^{-1}\mathbf{b}$, and observe that

$$(\mathbf{P}^{-1}\mathbf{A})^{t-1}\mathbf{P}^{-1}\mathbf{b} = (\mathbf{P}^{-1}\mathbf{A})\mathbf{v} = (\mathbf{I} + \mathbf{X})\mathbf{A}\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{X}\mathbf{A}\mathbf{v}.$$

Clearly $\mathbf{XAv} \subset \operatorname{range}(\mathbf{X}) \subseteq \mathbf{K}_{s+1}(\mathbf{A}, \mathbf{\Omega})$. Moreover, as noted, the vector $\mathbf{v}$ satisfies $\mathbf{v} \in \mathcal{K}_{t-1}(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s-1}(\mathbf{A}, \mathbf{\Omega})$, and so

$$\mathbf{A}\mathbf{v} \in \mathbf{A}\mathcal{K}_{t-1}(\mathbf{A}, \mathbf{b}) + \mathbf{A}\mathcal{K}_{t+s-1}(\mathbf{A}, \mathbf{\Omega}) \subseteq \mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \mathbf{\Omega}).$$

This proves the result. □

Theorem 3.2 allows to relate the performance of the augmented block-CG algorithm to the PCG one, with a broad choice of preconditioner.

THEOREM 3.3. *Fix any matrix $\mathbf{\Omega} \in \mathbb{R}^{d\times m}$, and let $\mathbf{P}_\mu = (\mathbf{I} + \mathbf{X})^{-1}$ be any preconditioner where* $\operatorname{range}(\mathbf{X}) \subseteq \mathcal{K}_{s+1}(\mathbf{A}, \mathbf{\Omega})$. *Define the augmented starting block $\mathbf{B} = [\mathbf{b}\ \mathbf{\Omega}]$. Then, for any $t \geq s$, the t-th block-CG iterate is related to the $(t-s)$-th preconditioned-CG iterate corresponding to the preconditioner $\mathbf{P}_\mu$ such that*

$$\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu} \leq \|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{pcg}_{t-s}(\mu)\|_{\mathbf{A}_\mu}.$$

Proof. Clearly $\mathcal{K}_t(\mathbf{A}, \mathbf{b}) + \mathcal{K}_{t+s}(\mathbf{A}, \mathbf{\Omega}) \subseteq \mathcal{K}_{t+s}(\mathbf{A}, [\mathbf{b}\ \mathbf{\Omega}])$, so the result follows immediately from Theorem 3.2, the optimality of the block-CG algorithm, and the definition of the preconditioned CG method. □

REMARK 3.4. Theorem 3.3 asserts that the augmented block-CG method automatically performs no worse than the Nyström PCG method (with the best choice of s and θ) after the same number of matrix loads.[5] In particular, when $\mathbf{UDU}^\mathsf{T} = \mathbf{A}\langle\mathbf{K}_s\rangle$, where $\langle\mathbf{K}_s\rangle$ is the Nyström approximation (2.3), then the deflation preconditioner $\mathbf{P}_\mu$ defined in (2.2) has the form

$$\mathbf{P}^{-1} = \mathbf{I} + \mathbf{X}, \qquad \text{where } \operatorname{range}(\mathbf{X}) \subseteq \mathcal{K}_{s+1}(\mathbf{A}, \mathbf{\Omega}).$$

This is a *deterministic* statement that explains the relative performance of the algorithms in Figure 1.1.

In Section 4, we use Theorem 3.3 and a new bound for Nyström preconditioning (Theorem 4.7), when $\mathbf{\Omega}$ is a Gaussian random matrix, to prove probabilistic bounds for the block-CG algorithm reminiscent of Corollary 2.4 for the spectral deflation preconditioner.

3.2. Computational costs.

The block-CG method is a standard algorithm in numerical analysis, and there are many mathematically equivalent implementations, i.e., different implementations that produce the same output in exact arithmetic. Here we describe a block-Lanczos-based implementation of the block-CG method that is particularly suitable for our applications to ridge regression and Gaussian sampling.

[5]Recall that s matrix loads are used to compute the Nyström approximation $\mathbf{A}\langle\mathbf{K}_s\rangle$, whose range lives in $\mathcal{K}_{s+1}(\mathbf{A}, \mathbf{\Omega})$.

The block-Lanczos algorithm applied to $(\mathbf{A},\mathbf{B})$ produces, after t iterations, matrices $\mathbf{Q}_t$ with orthonormal columns and $\mathbf{T}_t$ with bandwidth $2m+1$ satisfying

$$\operatorname{range}(\mathbf{Q}_t)=\mathcal{K}_t(\mathbf{A},\mathbf{B}),\qquad \mathbf{T}_t=\mathbf{Q}_t^{\mathsf{T}}\mathbf{A}\mathbf{Q}_t.$$

A standard computation reveals that

$$\mathsf{bcg}_k^{(1)}(\mu)=\|\mathbf{b}\|\mathbf{Q}_t(\mathbf{T}_t+\mu\mathbf{I})^{-1}\mathbf{e}_1, \tag{3.1}$$

where $\mathbf{e}_1=[1,0,\ldots,0]^{\mathsf{T}}$. In particular, since $\mathcal{K}_t(\mathbf{A}_\mu,\mathbf{B})=\mathcal{K}_t(\mathbf{A},\mathbf{B})$ for any scalar μ, (3.1) can be used to compute the block-CG iterate for *multiple values of* μ using the same $\mathbf{Q}_t$ and $\mathbf{T}_t$. For more details on the block-Lanczos method and the connection to the block-CG method, see, e.g., [8, 54, 62, 67].

There are *many* variants of the block-Lanczos method since there are many possible orthogonalization schemes that can be used within block-Lanczos-type algorithms [1, 7, 53]. The following provides a high-level overview of the costs associated with block-Lanczos methods with full reorthogonalization:

THEOREM 3.5. *Suppose that* $\mathbf{B}\in\mathbb{R}^{d\times m}$. *The block-Lanczos algorithm (with full orthogonalization) applied to* $(\mathbf{A},\mathbf{B})$ *produces, after t iterations, matrices* $\mathbf{Q}_t$ *and* $\mathbf{T}_t$ *using:*

- $t-1$ *matrix loads of* $\mathbf{A}$ *(for a total of* $m(t-1)$ *total matrix products),*
- $O(dm^2t^2)$ *floating-point operations (in addition to products with* $\mathbf{A}$*), and*
- $O(dmt)$ *storage (in addition to the storage required for* $\mathbf{A}$*).*

Subsequently, for any $\mu\ge 0$*, the t-th block-CG iterates* $\mathsf{bcg}_t^{(1)}(\mu)$ *can be computed using additional* $O(dmt+m^3t)$ *floating-point operations.*

Proof. Let $n=\dim(\mathcal{K}_t(\mathbf{A},\mathbf{B}))$. Since $\mathbf{B}$ has m columns, by definition (1.5) of the block Krylov subspace, $n\le\min\{d,mt\}$. At each iteration the block-Lanczos algorithm performs one matrix load (m parallel matrix products) and lower-order arithmetic. The dominant storage cost is to store $\mathbf{Q}_t$, which has $n\le mt$ columns of length n. The arithmetic cost is dominated by the $O(dn^2)=O(dm^2t^2)$ operations required to obtain $\mathbf{Q}_t$ using full reorthogonalization. Recall that $\mathbf{T}_t$ (and hence $\mathbf{T}_t+\mu\mathbf{I}$) is an $n\times n$ matrix with bandwidth $O(m)$. Thus, the linear system $(\mathbf{T}_t+\mu\mathbf{I})\mathbf{z}=\mathbf{e}_1$ can be solved in $O(nm^2)=O(m^3t)$ time. Subsequently, since $\mathbf{Q}_t$ is a $d\times n$ matrix, $\mathbf{Q}_t\mathbf{z}$ can be computed in $O(dn)=O(dmt)$ time. ∎

We emphasize that, similar to the unpreconditioned CG method (see, e.g., [5, 28, 39]), after the block-Lanczos algorithm has been run, the cost to compute the block-CG iterates for multiple values of μ is relatively small. This is in contrast to the Nyström PCG method, which is not particularly well suited for solving (1.1) for multiple values of μ: while the Nyström approximation can be reused, the PCG algorithm must be re-run, and new products with $\mathbf{A}$ have to be computed.

Notice that the costs (besides the matrix loads) scale with the block size m. We again remind the reader that, while Theorem 3.3 guarantees that the augmented block-CG method outperforms Nyström preconditioning in terms of matrix loads, our theory provides no guarantees for other costs such as matrix products or floating-point operations. Users should carefully consider the costs and benefits associated with the various algorithms in the context of their own computing environments.

REMARK 3.6. By avoiding orthogonalizing and storing the whole Krylov basis, it is in fact possible to reduce the floating-point costs to $O(dmtL+m^3tL)$ and the storage to $O(dm+dL)$, where L is the number of values of μ at which one wishes to evaluate $\mathsf{bcg}_t^{(1)}(\mu)$ [54]. However, there are several caveats to such an implementation. First, such methods require knowing the values of μ at which one wishes to evaluate $\mathsf{bcg}_t^{(1)}(\mu)$ ahead of time. Second, the finite-precision behavior of such methods can differ highly from the exact arithmetic behavior. We explore the impacts of finite-precision arithmetic in Appendix A, where we perform some numerical experiments that indicate that full reorthogonalization may be unnecessary in some situations.

For methods based on the standard Lanczos algorithm, there is quite a bit of theory that guarantees that such methods can still work, even without orthogonalization [8, 20, 32, 52, 55, 56]. Block Krylov methods must contend with additional difficulties such as blocks becoming ill-conditioned or even rank-deficient as the iteration proceeds (which can happen even in exact arithmetic). Unfortunately, much less theory is known about the block-Lanczos algorithm in finite-precision arithmetic [70]. Without reorthogonalization, we observe that in many cases the block-CG method fails to converge at all, while in other cases it does converge for some time; see Appendix A. Understanding this behavior is well beyond the scope of the current work.

4. Probabilistic bounds. In this section we prove probabilistic bounds for the augmented block-CG algorithm. In light of Theorem 3.3, our strategy for obtaining bounds for the block-CG method is simply to derive bounds for the Nyström PCG method. To do this, we follow the approach developed in [25] and use a deterministic bound for the condition number of the Nyström-preconditioned system. Note that the block-CG method always performs at least as well as CG (in terms of matrix loads) and hence is guaranteed to converge (in exact arithmetic).

The following is a minor generalization of [25, Proposition 5.3] to allow arbitrary θ. The proof is contained in Appendix C.1 for completeness.

PROPOSITION 4.1. *Let $\mathbf{P}_\mu$ be the Nyström preconditioner (2.2) corresponding to the Nyström approximation $\mathbf{A}\langle\mathbf{K}\rangle$ for any $\mathbf{K}$ and shift parameter $\theta \geq 0$. Then,*

$$\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}) \leq \left(\theta + \mu + \|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\|\right)\left(\frac{1}{\theta+\mu} + \frac{1}{\lambda_d + \mu}\right).$$

In particular, Proposition 4.1 implies that if $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\| \approx \lambda_{r+1}$ and $\theta \in [\lambda_d, \lambda_{r+1}]$, then $\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}) \approx \kappa_{r+1}(\mu)$.

Low-rank approximation is one of the most-studied problems in randomized numerical linear algebra, and many bounds have been developed. In the remainder of this section, we employ these bounds to obtain guarantees for the augmented block-CG algorithm.

Our first result, which we prove in Section 4.1, is a "numerical analysis"-style bound based on the error guarantees for a Nyström low-rank approximation in [68].

THEOREM 4.2. *Let $\mathbf{\Omega} \in \mathbb{R}^{d\times(r+p)q}$, where $q \geq \log(1/\delta)/\log(100)$, be a random Gaussian matrix. Suppose that $p \geq 2$ and*

$$s \geq \min\left\{3 + \frac{\log(d)}{2}, \frac{3}{2} + \frac{1}{4}\log\left(4 + \frac{4r}{p-1}\sum_{i>r}\frac{\lambda_i^2}{\lambda_{r+1}^2}\right)\right\}.$$

Then, the block-CG iterate $\mathsf{bcg}_t^{(1)}(\mu)$ satisfies, with probability at least $1-\delta$,

$$\left\{\forall\mu \geq 0 : \frac{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \leq 2\exp\left(-\frac{t-s}{3\sqrt{\kappa_{r+1}(\mu)}}\right)\right\}.$$

This implies that, after a small burn-in period of at most $O(\log(d))$ iterations, we have exponential convergence at a rate roughly $1/\sqrt{\kappa_{r+1}(\mu)}$.

Note that $\mathbf{A}\langle\mathbf{K}_s\rangle$, as defined in (2.3), can have a rank as large as ℓs, so we might hope that we can deflate roughly ℓs eigenvalues. Recent work [9, 50] implies that this is more or less the case. In Section 4.2 we use this result to prove our second result, a "theoretical computer science"-type bound.

THEOREM 4.3. *Let $\mathbf{\Omega} \in \mathbb{R}^{\ell}$ be a Gaussian matrix and $r \geq 0$. Then, for some*

$$s = O\left(\frac{r}{\ell}\log(\Delta) + \log\left(\frac{d}{\delta}\right)\right), \qquad \Delta := \min_{i=1,\ldots,\ell\lceil r/\ell\rceil-1} \frac{\lambda_i - \lambda_{i+1}}{\lambda_1},$$

the block-CG iterate $\mathsf{bcg}_t^{(1)}(\mu)$ satisfies, with probability at least $1-\delta$,

$$\left\{\forall \mu \geq 0 : \frac{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \leq 2\exp\left(-\frac{t-s}{3\sqrt{\kappa_{r+1}(\mu)}}\right)\right\}.$$

In contrast to Theorem 4.2, which requires $\ell \geq r+2$, Theorem 4.3 allows any choice of ℓ. The bound reveals that we obtain exponential convergence at a rate of roughly $1/\sqrt{\kappa_{r+1}(\mu)}$, after roughly $s \approx r/\ell$ matrix loads (and hence $r \approx \ell s$ matrix-vector products). Note that the logarithmic dependence on the eigenvalue gaps Δ is generally considered to be mild due to "smoothed-analysis"-type effects of finite-precision arithmetic [9, 50].

REMARK 4.4. The fact that Theorems 4.2 and 4.3 guarantee an accurate result for all $\mu \geq 0$ with high probability (as opposed to a single value of μ) will be important in our applications. In particular, it allows guarantees for computing the entire ridge-regression regularization path, and this will be necessary in the analysis of our algorithm for sampling Gaussian vectors.

REMARK 4.5. Bounds based on condition numbers (such as Theorem 4.2) are often pessimistic in practice. To determine how many iterations to run, it is more common to use some sort of a posteriori error estimate. For example, monitoring the residual $\|\mathbf{b} - \mathbf{A}_\mu \mathsf{bcg}_t^{(1)}(\mu)\|$ gives some indication of the quality of the solution. More advanced techniques can also be used [49].

4.1. Explicit bound.

In this section, we prove Theorem 4.2. We do not attempt to optimize constants, opting instead for simple arguments and clean theorem statements.

We begin by recalling an error guarantee for a Nyström low-rank approximation that compares the Nyström error $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}_s\rangle\|$ to the error $\|\mathbf{A} - [\![\mathbf{A}]\!]_r\| = \lambda_{r+1}$ of the best possible rank-r approximation to $\mathbf{A}$.

THEOREM 4.6 ([68, Theorem 9.1]). *Suppose that $\mathbf{\Omega} \in \mathbb{R}^{d\times(r+p)}$ is a random Gaussian matrix, and define $\mathbf{K}_s := [\mathbf{\Omega}\ \mathbf{A}\mathbf{\Omega}\ \cdots\ \mathbf{A}^{s-1}\mathbf{\Omega}]$. Then, if $p \geq 2$,*

$$\log\left(\frac{\mathbb{E}\,\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}_s\rangle\|^2}{\lambda_{r+1}^2}\right) \leq \frac{1}{8(s-\frac{3}{2})^2}\log\left(4 + \frac{4r}{p-1}\sum_{i>r}\frac{\lambda_i^2}{\lambda_{r+1}^2}\right)^2.$$

To prove our main convergence guarantee for our augmented block-CG algorithm, we derive a new bound for Nyström preconditioning that may be of independent interest.

THEOREM 4.7. *Suppose that $\mathbf{\Omega} \in \mathbb{R}^{d\times(r+p)}$ is a random Gaussian matrix, and define $\mathbf{K}_s := [\mathbf{\Omega}\ \mathbf{A}\mathbf{\Omega}\ \cdots\ \mathbf{A}^{s-1}\mathbf{\Omega}]$. Let $\mathbf{P}_\mu$ be the Nyström preconditioner (2.2) corresponding to the Nyström approximation $\mathbf{A}\langle\mathbf{K}_s\rangle$ for any shift parameter $\theta \in [\lambda_d, \lambda_{r+1}]$. Suppose that $p \geq 2$ and*

$$s \geq \min\left\{3 + \frac{\log(d)}{2}, \frac{3}{2} + \frac{1}{4}\log\left(4 + \frac{4r}{p-1}\sum_{i>r}\frac{\lambda_i^2}{\lambda_{r+1}^2}\right)\right\}.$$

Then, with probability at least $99/100$*,*

$$\left\{\forall\mu \geq 0 : \kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}) \leq 28\kappa_{r+1}(\mu)\right\}.$$

Note that our bound for the Nyström PCG method applies even when $\mu = 0$, whereas the bounds in [25] require μ to be sufficiently large relative to the sketching dimension $r + p$, and only $s = 1$ is considered.

Proof. The proof is a simple consequence of Theorem 4.6 and Proposition 4.1. We begin by bounding $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\|$. We first note that the minimum is always attained by the second term. Indeed, since $r \leq d$ and $\lambda_i/\lambda_{r+1} \leq 1$, for $i > r$, and since $p \geq 2$, we have

$$\frac{4r}{(p-1)}\sum_{i>r}\frac{\lambda_i^2}{\lambda_{r+1}^2} \leq 4d^2.$$

Now, using properties of the logarithm,

$$\log(4 + 4d^2) = \log(1 + d^2) + \log(4) \leq \log(d^2) + \log(2) + \log(4) = \log(8) + 2\log(d).$$

The claim then follows since $3/2 + \log(8)/4 \leq 3$.

With the condition on s, Theorem 4.6 guarantees that

$$\mathbb{E}\Big[\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\|^2\Big] \leq e^{1/2}\cdot\lambda_{r+1}^2.$$

Applying Markov's inequality, we therefore obtain that

$$\mathbb{P}\Big[\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\|^2 \geq 100\cdot e^{1/2}\lambda_{r+1}^2\Big] \leq \frac{1}{100}.$$

Now condition on the event that $\{\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\| \leq 10e^{1/4}\lambda_{r+1}\}$. Since $\theta \leq \lambda_{r+1}$ and $\mu \geq 0$, we have

$$\theta + \mu + \|\mathbf{E}\| \leq (1 + 10e^{1/4})(\lambda_{r+1} + \mu) \leq 14(\lambda_{r+1} + \mu).$$

Next, since $\theta \geq \lambda_d$,

$$\left(\frac{1}{\theta + \mu} + \frac{1}{\lambda_d + \mu}\right) \leq \frac{2}{\lambda_d + \mu}.$$

The result follows by combining the above equations. ◻

The proof of Theorem 4.2 is now straightforward. To get a high-probability bound, we use a simple boosting "trick". We expect that more refined results can be obtained directly.

Proof of Theorem 4.2. We first analyze the case $q = 1$ (i.e., $\delta = 1/100$). By Theorem 4.7, we are guaranteed that, with probability at least $99/100$,

$$\left\{\forall\mu \geq 0 : \kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}) \leq 28\kappa_{r+1}(\mu)\right\}. \tag{4.1}$$

When the event in (4.1) holds, Corollary 2.3 guarantees that

$$\left\{\forall\mu \geq 0 : \frac{\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{pcg}_t(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \leq 2\exp\left(-\frac{2t}{\sqrt{28\kappa_{r+1}(\mu)}}\right)\right\}.$$

The result follows from Theorem 3.3 and the fact that $\sqrt{28}/2 < 3$.

We now analyze the case for arbitrary $q \geq 1$. Partition $\mathbf{\Omega} = [\mathbf{\Omega}_1 \cdots \mathbf{\Omega}_q] \in \mathbb{R}^{d\times(r+2)q}$. Then, for $i = 1,\dots,q$, $\mathbf{\Omega}_i \in \mathbb{R}^{d\times(r+2)}$ are independent Gaussian matrices. By our analysis of the $q = 1$ case, the block-CG algorithm with starting block $[\mathbf{b}\ \mathbf{\Omega}_i]$ fails to reach the specified accuracy within t iterations with probability at most $1/100$. The probability that all of these q (independent) instances fail to reach this accuracy is therefore at most $(1/100)^q \leq \delta$. Finally, since

$$\mathcal{K}_t(\mathbf{A}, [\mathbf{b}\ \mathbf{\Omega}_i]) \subseteq \mathcal{K}_t(\mathbf{A}, \mathbf{\Omega}),$$

the block-CG algorithm with starting block $[\mathbf{b}\ \mathbf{\Omega}]$ performs no worse than the block-CG algorithm with starting block $[\mathbf{b}\ \mathbf{\Omega}_i]$ (for any i) and hence fails to reach the stated accuracy within t iterations with probability at most δ. ∎

4.2. Improved matrix-vector product complexity. In this section, we prove Theorem 4.3. The general approach is identical to the proof of Theorem 4.2, but we use a different bound for the Nyström low-rank approximation.

THEOREM 4.8 ([9, Theorem 1.3]). *Let $\mathbf{\Omega} \in \mathbb{R}^{d\times\ell}$ be a Gaussian matrix. Then, for some*

$$s = O\left(\frac{r}{\ell\sqrt{\varepsilon}}\log(\Delta) + \log\left(\frac{d}{\delta\varepsilon}\right)\right), \qquad \Delta := \min_{i=1,\dots,\ell\lceil r/\ell\rceil-1} \frac{\lambda_i - \lambda_{i+1}}{\lambda_1},$$

with probability at least $1-\delta$, there is a matrix $\mathbf{Q} \in \mathbb{R}^{d\times r}$ with orthonormal columns and $\operatorname{range}(\mathbf{Q}) \subseteq \mathcal{K}_s(\mathbf{A}, \mathbf{\Omega})$ *such that*

$$\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{A}\| < (1+\varepsilon)\lambda_{r+1}.$$

Note that Theorem 4.8 is not stated for a Nyström low-rank approximation. However, a Nyström low-rank approximation is better than a projection-based low-rank approximation [68], which allows us to prove Theorem 4.3.

Proof of Theorem 4.3. We set $\varepsilon = 10e^{1/4}-1$. Since $\operatorname{range}(\mathbf{Q}) \subseteq \operatorname{range}(\mathbf{K}_s)$, the standard monotonicity property of the Nyström approximation implies $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}_s\rangle\| \leq \|\mathbf{A} - \mathbf{A}\langle\mathbf{Q}\rangle\|$. Next, [68, Lemma 5.2] asserts that for any matrix $\mathbf{Q}$ with orthonormal columns, we obtain that $\|\mathbf{A} - \mathbf{A}\langle\mathbf{Q}\rangle\| \leq \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{A}\|$. Thus, by Theorem 4.8 (with ε set to an appropriate constant), the given condition on s ensures that, with probability at least $1-\delta$, the estimate $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}_s\rangle\| < 10e^{1/4}\lambda_{r+1}$. holds. The result then follows similarly to the proof of Theorem 4.2. ∎

REMARK 4.9. A number of algorithms for Gaussian-process regression make use of block-KSMs to simultaneously solve a linear system and apply a matrix function to a collection of Gaussian random vectors (for stochastic trace estimation) [30, 73]. Our bounds are relevant in this context and provide further theoretical justification for the use of block-KSMs in these settings.

5. Gaussian sampling.

Several applications in data science and statistics require sampling Gaussians with a given mean and covariance [2, 3, 66, 71]. A standard approach is to transform an isotropic Gaussian vector. Indeed, suppose that $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (i.e., the entries of $\mathbf{b}$ are independent standard Gaussians). Then, for positive definite $\mathbf{A}$,

$$\boldsymbol{\mu} + \mathbf{A}^{1/2}\mathbf{b} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}).$$

In other words, in order to sample Gaussian vectors with covariance $\mathbf{A}$, it suffices to apply the matrix square root of $\mathbf{A}$ to a standard Gaussian vector and then shift this result by $\boldsymbol{\mu}$. The most computationally difficult part of this is applying the square root of $\mathbf{A}$ to $\mathbf{b}$.

Recall the relation (1.3)

$$\mathbf{A}^{1/2}\mathbf{b} = \frac{2}{\pi}\int_0^\infty \mathbf{A}(\mathbf{A}+z^2\mathbf{I})^{-1}\mathbf{b}\,\mathrm{d}z.$$

When t is sufficiently large, we might hope that $\mathsf{cg}_t(z^2) \approx (\mathbf{A}+z^2\mathbf{I})^{-1}\mathbf{b}$. This motivates the following definition:

DEFINITION 5.1. *The t-th Lanczos square root iteration is defined as*

$$\mathsf{sq}_t := \frac{2}{\pi}\int_0^\infty \mathbf{A}\mathsf{cg}_t(z^2)\,\mathrm{d}z.$$

This and closely related methods appear throughout the literature [8, 13, 59].

Often we wish to sample multiple vectors from $\mathcal{N}(\boldsymbol{\mu},\mathbf{A})$, and so we might use a block variant of Definition 5.1.

DEFINITION 5.2. *Let $\mathbf{B} = [\mathbf{b}^{(1)}\cdots\mathbf{b}^{(m)}]$. The t-th block-Lanczos square root iterate is defined as*

$$\mathsf{bsq}_t^{(i)} := \frac{2}{\pi}\int_0^\infty \mathbf{A}\mathsf{bcg}_t^{(i)}(z^2)\,\mathrm{d}z.$$

Both algorithms can be efficiently implemented using the (block)-Lanczos algorithms; see, e.g., [8].

We can use the bounds for the augmented block-CG algorithm from Section 4 to derive bounds for the block-Lanczos square root algorithm. In particular, using Theorem 4.2, we prove the following:

THEOREM 5.3. *Let $\mathbf{b}^{(1)},\ldots,\mathbf{b}^{(m)} \in \mathbb{R}^d$ be independent standard Gaussian random vectors. Suppose that*

$$r \le (m-1)\left\lceil \frac{\log(m/\delta)}{\log(100)} \right\rceil^{-1} - 2.$$

Then, with probability at least $1-\delta$,

$$\left\{\forall i : \frac{\|\mathbf{A}^{1/2}\mathbf{b}^{(i)} - \mathsf{bsq}_t^{(i)}\|}{\|\mathbf{A}^{1/2}\|\|\mathbf{b}^{(i)}\|} \le \log(16\kappa)\exp\left(-\frac{t-(2+\log(d)/2)}{3\sqrt{\kappa_{r+1}(0)}}\right)\right\}.$$

Similarly, using Theorem 4.3, we obtain the next theorem.

THEOREM 5.4. *Let $\mathbf{b}^{(1)},\ldots,\mathbf{b}^{(m)} \in \mathbb{R}^d$ be independent standard Gaussian random vectors and $r \ge 0$. Then, for some*

$$s = O\left(\frac{r}{m}\log(\Delta) + \log\left(\frac{d}{\delta}\right)\right), \qquad \Delta := \min_{i=1,\ldots,(m-1)\lceil r/(m-1)\rceil-1} \frac{\lambda_i - \lambda_{i+1}}{\lambda_1},$$

with probability at least $1-\delta$,

$$\left\{\forall i : \frac{\|\mathbf{A}^{1/2}\mathbf{b}^{(i)} - \mathsf{bsq}_t^{(i)}\|}{\|\mathbf{A}^{1/2}\|\|\mathbf{b}^{(i)}\|} \le \log(16\kappa)\exp\left(-\frac{t-s}{3\sqrt{\kappa_{r+1}(0)}}\right)\right\}.$$

Proof of Theorems 5.3 and 5.4. Our proof is optimized for simplicity rather than sharpness. For notational convenience, let $\mathbf{b} = \mathbf{b}^{(1)}$ and $\boldsymbol{\Omega} = [\mathbf{b}^{(2)},\ldots,\mathbf{b}^{(m)}]$. As noted in Section 5, $\boldsymbol{\Omega} \in \mathbb{R}^{d\times(m-1)}$ is independent of $\mathbf{b}$. Consider the event

$$\left\{\forall \mu \ge 0 : \frac{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \le \varepsilon_t'(\mu)\right\}, \tag{5.1}$$

where

$$\varepsilon_t'(\mu) = 2\exp\left(-\frac{t-s}{3\sqrt{\kappa_{r+1}(\mu)}}\right).$$

To prove Theorem 5.3, note that the choice of r ensures that $m-1 \geq (r+2)q$ for $q = \lceil \log(m/\delta)/\log(100)\rceil$. Therefore, with $s = 2+\log(d)/2$, Theorem 4.2 guarantees that (5.1) holds with probability at least $1-\delta/m$. Likewise, to prove Theorem 5.4, note that Theorem 4.3 guarantees that (5.1) holds with probability at least $1-\delta/m$ (where we have used that $m \leq d$ so that $\log(dm/\delta) = O(\log(d/\delta))$).

From this point on, we condition on (5.1), and the two proofs are identical. Applying standard norm inequalities, we obtain a bound

$$\begin{aligned}\|\mathbf{A}\mathbf{A}_\mu^{-1}\mathbf{b} - \mathbf{A}\mathsf{bcg}_t^{(1)}(\mu)\| &= \|\mathbf{A}\mathbf{A}_\mu^{-1/2}\mathbf{A}_\mu^{1/2}(\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu))\| \\ &\leq \|\mathbf{A}\mathbf{A}_\mu^{-1/2}\|\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu}.\end{aligned}$$

Under the assumption that the event in (5.1) holds, we bound

$$\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu} \leq \varepsilon_t(\mu)\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu} \leq \varepsilon_t(\mu)\|\mathbf{A}_\mu^{-1/2}\|\|\mathbf{b}\|.$$

We observe that

$$\|\mathbf{A}\mathbf{A}_\mu^{-1/2}\| = \max_i \frac{\lambda_i}{\sqrt{\lambda_i+\mu}} = \frac{\lambda_1}{\sqrt{\lambda_1+\mu}}$$

and similarly,

$$\|\mathbf{A}_\mu^{-1/2}\| = \max_i \frac{1}{\sqrt{\lambda_i+\mu}} = \frac{1}{\sqrt{\lambda_d+\mu}}.$$

Therefore, since $\varepsilon_t'(\mu) \leq \varepsilon_t'(0)$,

$$\|\mathbf{A}\mathbf{A}_\mu^{-1}\mathbf{b} - \mathbf{A}\mathsf{bcg}_t^{(1)}(\mu)\| \leq \frac{\lambda_1\varepsilon_t'(0)\|\mathbf{b}\|}{\sqrt{(\lambda_1+\mu)(\lambda_d+\mu)}}. \tag{5.2}$$

Applying the triangle inequality for integrals and (5.2), we find

$$\begin{aligned}\|\mathbf{A}^{1/2}\mathbf{b} - \mathsf{bsq}_t^{(1)}\| &= \frac{2}{\pi}\left\|\int_0^\infty \mathbf{A}(\mathbf{A}+z^2\mathbf{I})^{-1}\mathbf{b} - \mathbf{A}\mathsf{bcg}_t^{(1)}(z^2)\mathrm{d}z\right\| \\ &\leq \frac{2}{\pi}\int_0^\infty \|\mathbf{A}(\mathbf{A}+z^2\mathbf{I})^{-1}\mathbf{b} - \mathbf{A}\mathsf{bcg}_t^{(1)}(z^2)\|\mathrm{d}z \\ &\leq \frac{2}{\pi}\int_0^\infty \frac{\lambda_1\varepsilon_t'(0)\|\mathbf{b}\|}{\sqrt{(\lambda_1+z^2)(\lambda_d+z^2)}}\mathrm{d}z. \qquad (5.3)\end{aligned}$$

A direct computation reveals

$$\int_0^\infty \frac{1}{\sqrt{(\lambda_1+z^2)(\lambda_d+z^2)}}\mathrm{d}z = \frac{K(1-\lambda_1/\lambda_d)}{\sqrt{\lambda_d}}, \tag{5.4}$$

where $K(m) := \int_0^{\pi/2}(1-m\sin^2(z))^{-1/2}\mathrm{d}z$ is the complete elliptic integral of the first kind. Standard bounds for elliptic integrals (see Lemma C.1) guarantee that, for all $x > 1$,

$$\frac{2}{\pi}\sqrt{x}K(1-x) \leq \frac{5}{4\pi}\log(16x) \leq \frac{1}{2}\log(16x). \tag{5.5}$$

Therefore, applying (5.5) with $x = \kappa = \lambda_1/\lambda_d$ to (5.3) and (5.4), we get a bound

$$\|\mathbf{A}^{1/2}\mathbf{b} - \mathsf{bsq}_t^{(1)}\| \leq \frac{\sqrt{\lambda_1}}{2} \log\left(16\kappa\right) \varepsilon_t'(0) \|\mathbf{b}\|.$$

Since $\sqrt{\lambda_1} = \|\mathbf{A}^{1/2}\|$, this is the desired result for $\mathbf{b}^{(1)}$.

To obtain the final result, we observe that $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m)}$ and $\mathsf{bsq}_t^{(1)}, \ldots, \mathsf{bsq}_t^{(m)}$ are permutation invariant in distribution. Thus, the above result in fact applies to each $\mathbf{b}^{(i)}$ and $\mathsf{bsq}_t^{(i)}$ pair with probability at least $1 - \delta/m$. Applying a union bound over these m events gives the main result. □

Theorems 5.3 and 5.4 give bounds for the matrix-vector products required by the block-Lanczos method. For instance, Theorem 5.3 implies that we can roughly use

$$m\Big(\log(d) + \sqrt{\kappa_{r+1}} \log\left(\log(\kappa)/\varepsilon\right)\Big)$$

matrix-vector products, where $r = O(m/\log(m))$. In contrast, existing bounds for methods like the single-vector Lanczos square root method (Definition 5.1) [10, 59] sample a single Gaussian vector with roughly $\sqrt{\kappa}\log(1/\varepsilon)$ matrix-vector products. Therefore, when $\lambda_{r+1} \ll \lambda_1$, the total number of matrix-vector products is reduced significantly by using the block method in Definition 5.2.

REMARK 5.5. For sampling Gaussians, some preconditioning-like approaches can be applied [13, 29, 59]. As far as we are aware, there are no theoretical guarantees for such methods similar to those presented in the current work. In addition, analogous to the case of linear systems, we expect block methods to enjoy the benefits of working over a larger subspace.

REMARK 5.6. Relating KSMs for matrix functions to KSMs for (shifted) linear systems via integral relations is widely used in order to design and analyze algorithms for matrix functions; see, e.g., [10, 27, 64, 72]. However, it is generally difficult to use preconditioners for linear systems for matrix functions, as preconditioned Krylov subspaces do not generally satisfy a shift-invariance property. As such, the integral relation no longer yields an efficient-to-implement algorithm. On the other hand, we expect similar analyses, based on implicit preconditioning with the block-CG algorithm, to work for other functions.

5.1. Sampling with the inverse covariance. Computing $\mathbf{A}^{-1/2}\mathbf{b}$ is used to transform vectors with covariance matrix $\mathbf{A}$ to a "whitened" coordinate space, where the covariance is the identity, and this has found use in a number of data-science applications [42, 59].

We can define an approximation for applying inverse square roots similar to the block-Lanczos square root iterate (Definition 5.2).

DEFINITION 5.7. *Let* $\mathbf{B} = [\mathbf{b}^{(1)} \cdots \mathbf{b}^{(m)}]$. *The t-th block-Lanczos inverse square root iteration is defined as*

$$\mathsf{bisq}_t^{(i)} := \frac{2}{\pi} \int_0^\infty \mathsf{bcg}_t^{(i)}(z^2)\, \mathrm{d}z.$$

Note that $\mathsf{bisq}_t^{(i)} = \mathbf{A}^{-1}\mathsf{bsq}_t^{(i)}$. Therefore, Theorem 5.3 immediately gives a bound for Definition 5.7 since

$$\begin{aligned}\|\mathbf{A}^{-1/2}\mathbf{B} - \mathsf{bisq}_t\| &= \|\mathbf{A}^{-1/2}\mathbf{B} - \mathbf{A}^{-1}\mathsf{bsq}_t\| \\ &= \|\mathbf{A}^{-1}(\mathbf{A}^{1/2}\mathbf{B} - \mathsf{bsq}_t)\| \leq \|\mathbf{A}^{-1}\|\|\mathbf{A}^{1/2}\mathbf{B} - \mathsf{bsq}_t\|.\end{aligned}$$

6. Discussion and comparison with previous work.

There are many other related approaches to solving the problems in Section 1. For instance, KSMs are commonly used to solve Tikhonov regression problems, using the shift-invariance of Krylov subspaces (1.4) to efficiently solve for the whole regularization path [5, 28, 39]. Similarly, as discussed in Section 5, KSMs are also used for tasks involving matrix functions, including sampling Gaussian vectors [8, 13, 59]. The techniques discussed in Section 5 are likely applicable to other functions as well.

The block-CG algorithm is widely used to solve linear systems with multiple right-hand sides, and in such settings, working over the block Krylov subspace is often advantageous compared to working over the individual Krylov subspaces [4, 24, 47, 54]. The block-CG approach is also used to solve single linear systems. For instance, so-called enlarged KSMs split the right-hand size vector $\mathbf{b}$ into multiple vectors using a domain decomposition approach, and then apply the block-CG algorithm to this collection of vectors [34, 35, 36]. Augmenting a KSM with random vectors was shown to be beneficial in numerical experiments appearing in concurrent work [75, Appendix B].

While we are unaware of any bounds similar to Theorem 4.3, bounds similar to Theorem 4.2 are known. In particular, in [54, Section 4] a convergence bound for the block-CG algorithm is derived in terms of $\sqrt{\lambda_1/\lambda_{d-m+1}}$, i.e., the condition number of $\mathbf{A}$ with the *bottom* m eigenvalues removed.[6] At a high level, in [54] this bound is obtained by showing that the block Krylov subspace contains vectors that can approximately annihilate a subset of the eigenvalues of $\mathbf{A}$. Thus, in principle, other sets of eigenvalues (besides the bottom ones) can be eliminated. The bounds of [54] work for any starting block (but, as a result, are more technical, depending on how "good" the starting block is). It would not be surprising if such quantities can be bounded when the starting block is Gaussian.

KSMs are widely used for both low-rank approximation and problems involving matrix functions, but their behavior for these problems is fundamentally different. As such, it would be interesting to try to avoid relying on a black-box reduction for low-rank approximation. For example, while low-rank approximation is complicated by the presence of repeated eigenvalues, solving linear systems is not. The approach of [54] is closely related to the approaches used in [51, 68] to prove bounds for low-rank approximation, so it is conceivable that, by taking the approach of [54] but using modern tools from randomized numerical linear algebra, one may be able to directly derive better bounds for randomized versions of the block-CG method.

More broadly, there are many randomized algorithms for linear systems and regression. For instance, a recent line of work shows that certain stochastic iterative methods enjoy an implicit preconditioning-like effect [15, 17, 18, 19, 44], with convergence independent of the top eigenvalues. These algorithms have runtime guarantees better than KSMs in some settings [17]. In addition, for tall least-squares problems, the sketch-and-precondition framework for building a preconditioner for KSMs is extremely effective [46, 61], resulting in algorithms with optimal theoretical complexity [12, 14]. A number of recent works in randomized numerical linear algebra [9, 11, 50, 68] have made use of various nestedness properties of block-KSMs similar in flavor to Theorem 3.2 in order to obtain more efficient algorithms.

7. Numerical experiments.

We implement the block-CG and the block-Lanczos square root iterates using the block-Lanczos algorithm with full reorthogonalization in Python. The code to replicate the experiments in this paper is available at the repository `https://github.com/tchen-research/precond_without_precond`. The s matrix loads required to build the Nyström preconditioner are accounted for in the plots. For

[6] Note that the ordering of the eigenvalues in [54] is reversed from the present paper.

the Nyström PCG algorithm, we set θ as the smallest eigenvalue of the Nyström approximation. We use the same random Gaussian matrix $\mathbf{\Omega} \in \mathbb{R}^{d\times\ell}$ for the Nyström PCG and the block-CG method. More numerical experiments, including ones without full reorthogonalization, are included in Appendices B and A.

7.1. Convergence. We begin by comparing the convergence of the methods discussed in this paper for several test problems. The results of this experiment are illustrated in Figure 7.1, and, as expected, our augmented block-CG method outperforms the others, often by orders of magnitude. We also observe that the Nyström PCG method benefits from using $s > 1$, i.e., from performing more than one pass over $\mathbf{A}$ when building the preconditioner.

As discussed in Section 1.2, throughout this paper we assume that matrix loads (iterations) are the dominant cost. For reference, we have also displayed the convergence as a function of matrix-vector products in Figure A.4.

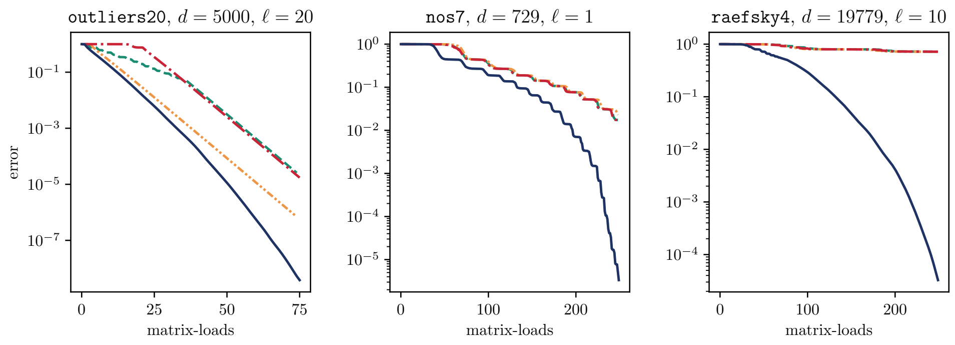


FIG. 7.1. *The relative error $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}}/\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ versus the matrix loads for the block-CG method (━━), the CG method (━·━), and the Nyström PCG method with $s = 1$ (━ ━ ━) and $s = 3$ (━··━) for several test problems.*

7.1.1. The introductory figure. Figure 1.1 is run for the `fastdecay` problem ($d = 8000$) with $\ell = 10$. Nyström preconditioning is run with a grid of parameters. Specifically, we use depths $s \in \{1, 5, 11\}$ and a range of θ, including the choice $\theta = \lambda_\ell(\mathbf{A}\langle\mathbf{\Omega}\rangle)$.

The timings are produced based on a setting where $\mathbf{A}$ cannot be stored in its entirety in fast memory. In particular, we store $\mathbf{A}$ in 8 separate 1000×8000 chunks and perform (block) matrix-vector products with $\mathbf{A}$ by sequentially loading a single chunk from the disk into the random-access memory and performing the appropriate part of the products. Similar to [68, Section 6.1], we observe that the runtime of the algorithm is dominated by the cost of loading these chunks into random-access memory.

7.2. Regularization parameter. In Figure 7.2 we display the error after a fixed number of matrix loads as a function of μ. As expected, this plot indicates that our augmented block-CG outperforms the other methods for each value of μ. Perhaps more importantly, our block-CG method can efficiently compute the solution to (1.1) for many values of μ, *without the need for additional matrix loads* (see Theorem 3.5). This is in contrast to the Nyström PCG method, which requires a new run for each value of μ. For tasks such as ridge regression, this is of note.

We note that the maximum attainable accuracy of the block-CG algorithm (an effect of performing the numerical experiments in finite-precision arithmetic, which prevents convergence to the true solution) seems to be higher than that of other methods, causing stagnation.

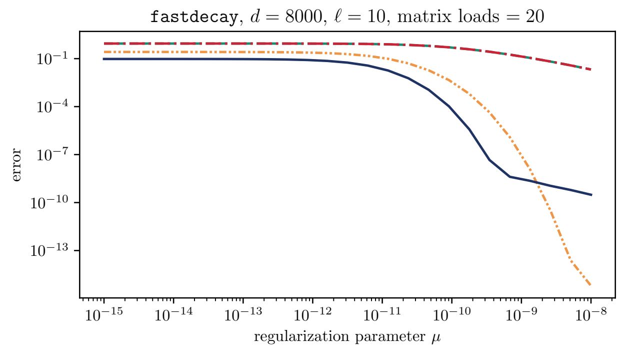


FIG. 7.2. *The relative error* $\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}_\mu} / \|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}$ *after a fixed number of matrix loads as a function of the regularization parameter* μ *for the block-CG method (* ━━ *), the CG method (* ━·━ *), and the Nyström PCG method with* $s = 1$ *(* ━ ━ ━ *) and* $s = 3$ *(* ━··━ *).*

7.3. Sampling Gaussians. We perform an experiment comparing the block-Lanczos square root iterate to the standard Lanczos square root iterate. In particular, in Figure 7.3, we compare the maximum error in approximating $\mathbf{A}^{1/2}\mathbf{b}^{(i)}$, $i = 1, \ldots, m$, for the two methods. All runs of the single-vector method are done in parallel so that the number of matrix loads is independent of m. The block method outperforms the single-vector method, and the performance gains are often significant.

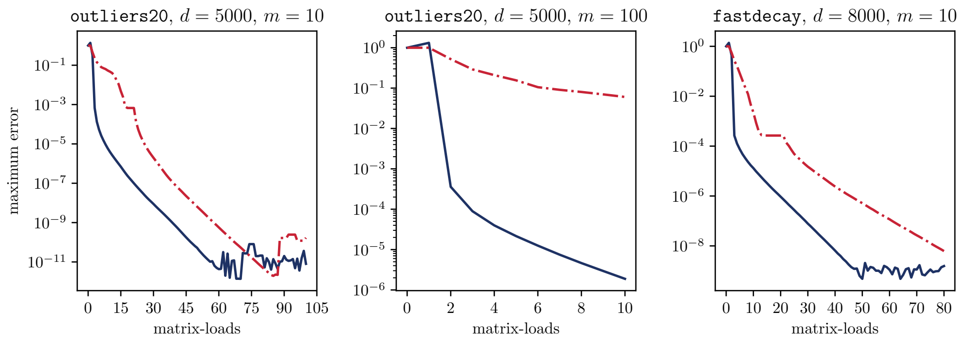


FIG. 7.3. *The maximum relative sample error* $\max_i \|\mathbf{A}^{1/2}\mathbf{b}_i - \mathsf{alg}\| / \|\mathbf{A}^{1/2}\mathbf{b}_i\|$ *versus the matrix loads for the Lanczos square root (* ━·━ *) and the block-Lanczos square root algorithm (* ━━ *).*

8. Outlook.

We have introduced a variant of the block-CG method which outperforms the Nyström PCG method in certain settings. Our work provides theoretical evidence of the virtues of block Krylov subspace methods for solving a single linear system of equations and for sampling Gaussian vectors. We believe that future work should study a more practical variant of the augmented block-CG algorithm, where deflation is used to reduce the block size as soon as it is recognized that the benefits of a large block size are no longer justified.

It would also be interesting to understand the extent to which similar ideas can be extended to other tasks involving matrix functions, such as estimating the trace of matrix functions. Existing algorithms often involve applying matrix functions to many Gaussian vectors [23, 50, 69], so it seems likely that using block-Krylov methods may be advantageous compared to using single-vector methods. However, existing theory does not reflect this.

Acknowledgements. We thank Ethan Epperly, Daniel Kressner, and David Persson for useful discussions and feedback.

Appendix A. More numerical experiments.

In this section we offer further numerical experiments which provide additional insight into the behavior of our augmented block-CG as well as the Nyström PCG algorithms.

A.1. Convergence.

We provide more test problems comparing the convergence of the block-CG with the Nyström PCG and the CG method, as described in Section 7.1. As observed in Figure 7.1, the block-CG algorithm outperforms the other methods in terms of matrix loads, and it often does so significantly.

Note that for the `outliers20` problem, we now append $\ell = 10$ or $\ell = 22$ Gaussian vectors (rather than $\ell = 20$ as shown in Figure 7.1). This problem has 20 large eigenvalues, and the small amount of oversampling significantly improves the convergence of the Nyström PCG method with $s = 1$. We also include a test with the `bottom20` problem, which has 20 small eigenvalues.

A.2. Block size.

Theorem 4.3 suggests that the accuracy of low-rank approximations of $\mathbf{A}$, built from the information in the block Krylov subspace $\mathcal{K}_t(\mathbf{A}, \mathbf{\Omega})$, where $\mathbf{\Omega}$ is a random Gaussian matrix, depends on bk, regardless of the individual values of b and k. In Figure A.2, we present the error (after a fixed number of matrix loads) as a function of the block size, and in Figure A.3, we display the condition number of $\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}$ as a function of the block size. In some cases, the Nyström PCG method with depth $s = 3$ behaves similarly to the Nyström PCG method with depth $s = 1$ when the same number of matrix-vector products is used. For instance, for the `outliers20` problem, both methods have a significant drop in the error (due to a significant drop in the condition number) when the dimension of the Krylov subspace is roughly equal to 20, the number of outlying eigenvalues.

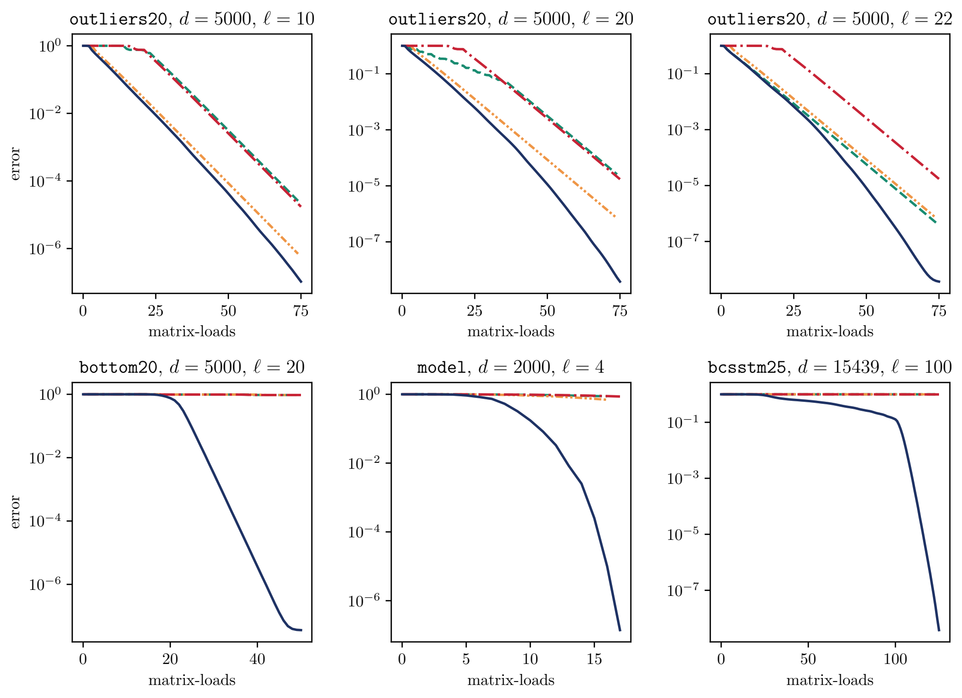


FIG. A.1. *The relative error $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}}/\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ versus the matrix loads for the block-CG method ( ), the CG method ( ), and the Nyström PCG method with $s = 1$ ( ) and $s = 3$ ( ) for several test problems; see also Figure 7.1.*

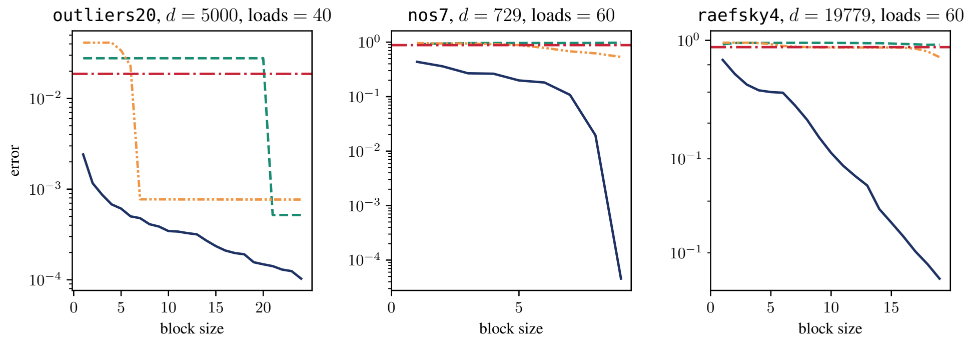


FIG. A.2. *The relative error $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}}/\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ versus the Nyström starting block size ℓ (the number of columns in $\mathbf{\Omega}$) for the block-CG method ( ), the CG method ( ), and the Nyström PCG method with $s = 1$ ( ) and $s = 3$ ( ) for several test problems.*

A.3. Matrix-vector cost. In Figure A.4, we display the convergence as a function of matrix-vector products for the same test problems as in Figure 7.1. Here, our block-CG algorithm underperforms the other methods. This does not conflict with any of our theory, which is stated in terms of matrix loads. However, it serves as a reminder that our method may not be suitable in all computational settings; see Section 1.2.

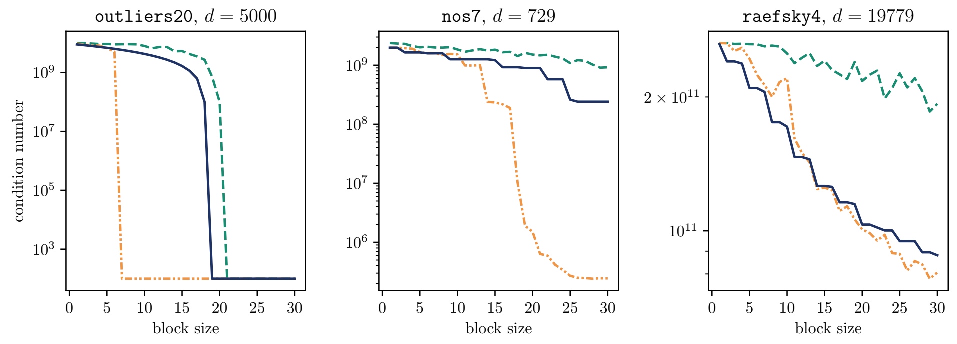


FIG. A.3. *The condition number $\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})$ of the Nyström preconditioner for $s = 1$ (- - -) and $s = 3$ (-··-) as a function of the block size ℓ (number of columns in $\mathbf{\Omega}$). For reference, we also show $(\lambda_{\ell+1} + \mu)/(\lambda_d + \mu)$ (——).*

A.4. Test problems. In Figure A.5, we display the spectrums of the test problems that we used.

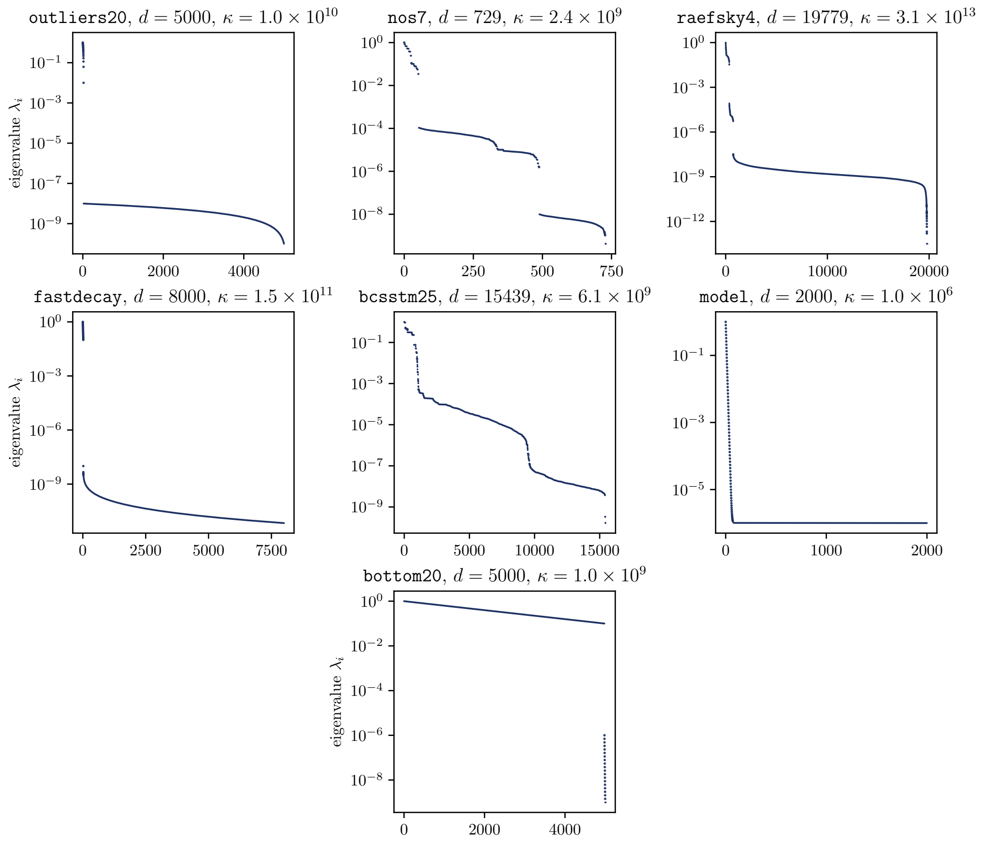


FIG. A.5. *Spectrum of test problems used in this paper.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

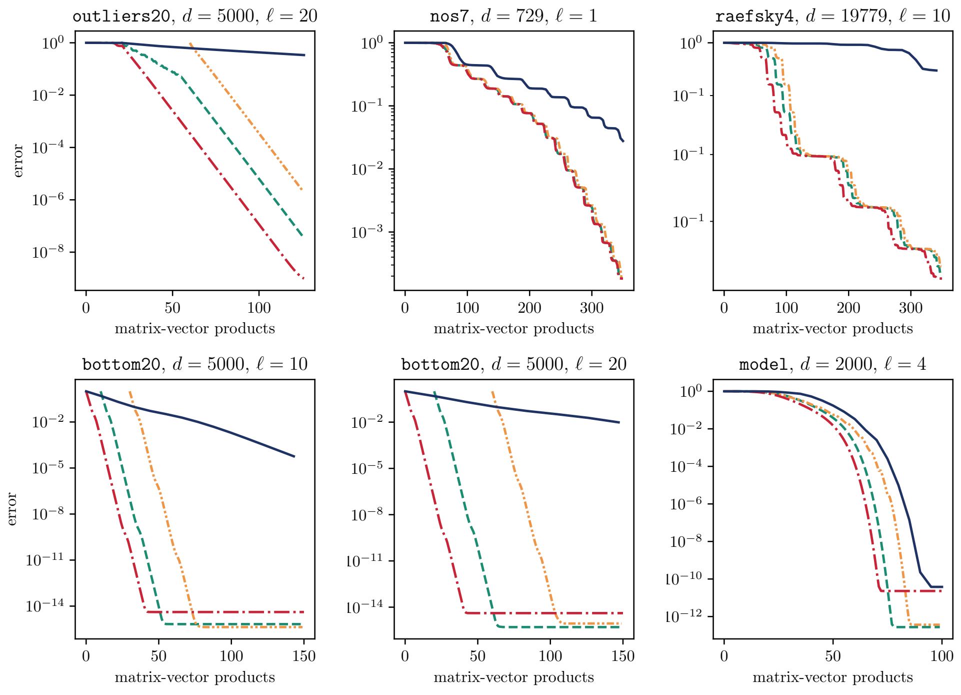


FIG. A.4. *The relative error* $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}}/\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ *versus the matrix-vector products for the block-CG method* (—), *the CG method* (–·–), *and the Nyström PCG method with* $s = 1$ (– – –) *and* $s = 3$ (–··–) *for the same test problems as in Figure* 7.1.

Appendix B. Further discussion of the Nyström PCG/CG/Block-CG method. Current theory for the Nyström PCG and CG methods raises some interesting questions about the efficacy of the Nyström PCG algorithm if costs are measured in terms of matrix-vector products. In particular, the bounds for the Nyström PCG method do not guarantee that the algorithm uses any fewer matrix-vector products than the CG method. In this section, we provide a discussion with the aim of raising some considerations about the Nyström PCG algorithm that are likely of importance to practitioners and may constitute interesting directions for future work.

B.1. Bounds for the CG method in terms of the deflated condition number. In Theorem 4.7, we show that if $s = O(\log(d))$, then $\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}) = O(\lambda_{r+1}/\lambda_d)$. It is informative to compare such a bound against similar bounds for the CG method. Towards this end, we recall a standard bound for the CG iterates; see, e.g., [33].

THEOREM B.1. *For any* $r \geq 0$*, the CG iterate satisfies*

$$\frac{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{cg}_t(\mu)\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b}\|_{\mathbf{A}_\mu}} \leq 2\exp\left(-\frac{2(t-r)}{\sqrt{\kappa_{r+1}(\mu)}}\right).$$

Theorem B.1 guarantees that the CG method converges at a rate of $\sqrt{\lambda_{r+1}/\lambda_d}$ after a burn-in period of r iterations. This is reminiscent of the bounds for the Nyström PCG method presented in this paper, which guarantee convergence at this rate, after a preconditioner build-time of roughly $O(r)$ matrix products.

B.2. Bounds in terms of the effective dimension. The main theoretical bounds for the Nyström PCG method from [25] are in terms of the effective dimension

$$d_{\text{eff}}(\mu) := \operatorname{tr}\big(\mathbf{A}\mathbf{A}_\mu^{-1}\big) = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i + \mu}.$$

The main result is a guarantee for the condition number of the Nyström-preconditioned system.

THEOREM B.2 ([25, Theorem 1.1]). *Let $\mathbf{\Omega} \in \mathbb{R}^{d\times\ell}$ be a matrix of independent standard normal random variables for some $\ell \geq 2\lceil 1.5 d_{\text{eff}}(\mu)\rceil + 1$. Then, with $\theta = \lambda_\ell(\mathbf{A}\langle\mathbf{\Omega}\rangle)$, the Nyström-preconditioned system satisfies*

$$\mathbb{E}\Big[\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})\Big] < 28.$$

When $\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})$ is bounded by a constant, the preconditioned CG method will converge to a fixed accuracy (e.g., 10^{-4} or 10^{-16}) in a number of iterations *independent* of the condition number or the spectral properties of $\mathbf{A}$.

We can convert Theorem B.1 to a bound in terms of the effective dimension by noting that all but the first $O(d_{\text{eff}})$ eigenvalues of $\mathbf{A}$ are bounded by μ. In particular, [25, Lemma 5.4] implies that for any r and $\mu \geq 0$,

$$r > 2d_{\text{eff}}(\mu) \quad\Longrightarrow\quad \lambda_{r+1} \leq \mu. \tag{B.1}$$

Together, Theorem B.1 and (B.1) give a bound for the CG method in terms of the effective dimension.

COROLLARY B.3. *The CG iterate satisfies*

$$\frac{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{cg}_t\|_{\mathbf{A}_\mu}}{\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathsf{cg}_0\|_{\mathbf{A}_\mu}} \leq 2\exp\left(-\sqrt{2}(t - 2d_{\text{eff}}(\mu))\right).$$

Corollary B.3 asserts that after a burn-in of $2d_{\text{eff}}(\mu)$ iterations, the CG method converges in $O(\log(1/\varepsilon))$ iterations. Thus, the theory in [25] does not guarantee that the Nyström PCG method has any major benefits over CG in terms of matrix-vector products.

B.3. Is Nyström PCG a good idea? The existence of a bound like Theorem B.1 seems damning; the CG method automatically satisfies bounds similar to those of the Nyström PCG method, without the need to construct and store a preconditioner. However, as we now discuss, the full story is much more subtle, and we believe the Nyström PCG algorithm is still a viable method in many instances.

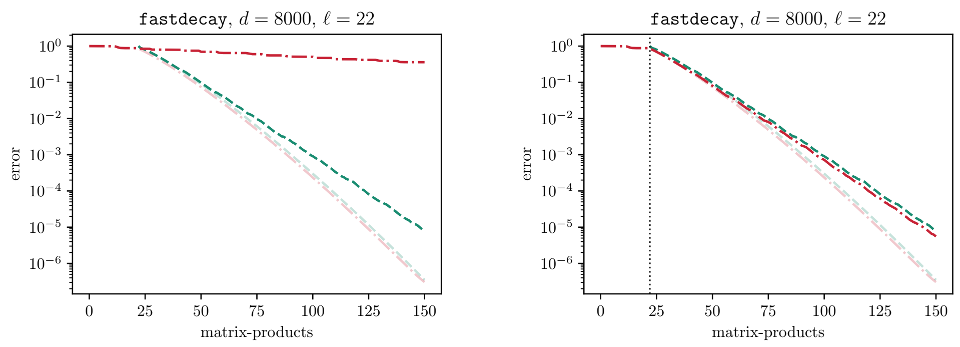
First, the Nyström PCG method is able to parallelize matrix-vector products used to build the preconditioner. Thus, the number of matrix loads can be considerably less than required by the CG method. This is reflected in our experiments in Figure 7.1.

However, these same plots indicate that in terms of matrix products, CG significantly outperforms the Nyström PCG method (the $s\ell$ matrix products to build the Nyström preconditioner require only one matrix load). On the other hand, in apparent contradiction to our experiments, the experiments in [25] indicate that the Nyström PCG method consistently and significantly outperforms the CG method.

Recall that the experiments in Section 7 are done using full reorthogonalization (so that exact arithmetic theory is still applicable). On the other hand, the experiments in [25] appear to have been done without reorthogonalization. It is well known that Theorem B.1 does not hold in finite-precision arithmetic without reorthogonalization for any $r > 0$.[7] On the other

[7] A weaker version does still hold [32].

hand, Corollary 2.3 can be expected to hold to a close degree [32, 48]. Since the Nyström PCG algorithm works by explicitly decreasing the condition number of $\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})$, *the convergence of the Nyström PCG method in finite-precision arithmetic is still well described by the exact arithmetic theory for Nyström PCG.*



(a) The CG method (— · —) and the Nyström PCG method with $s = 1$ (– – –) without any reorthogonalization.

(b) The CG method (— · —) with reorthogonalization for 22 iterations and the Nyström PCG method with $s = 1$ (– – –) without any reorthogonalization.

FIG. B.1. *The relative error* $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_\mathbf{A}/\|\mathbf{A}^{-1}\mathbf{b}\|_\mathbf{A}$ *as a function of the matrix products. The light curves show the convergence with full reorthogonalization.*

We illustrate this point of comparison explicitly in Figure B.1a. We run the CG and the Nyström PCG method for the `fastdecay` problem (for which $\lambda_1/\lambda_d \gg \lambda_{20}/\lambda_d$). As expected, the CG method with reorthogonalization burns in for roughly 20 iterations, and then begins converging much more rapidly at a rate depending on λ_{21}/λ_d, outperforming the Nyström PCG method with reorthogonalization ($s = 1, \ell = 22$), which uses 22 matrix-vector products[8] to build the preconditioner, at which point it also converges at a rate depending on λ_{21}/λ_d. However, if reorthogonalization is not used, then the CG method converges at a much slower rate depending on λ_1/λ_d while the Nyström PCG method still converges at the faster rate depending on λ_{21}/λ_d.

B.4. What if CG uses some reorthogonalization? Comparing the vanilla CG with the Nyström PCG method as in Figure B.1a is perhaps a bit unfair to CG; the Nyström PCG method incurs costs in building, storing, and applying the preconditioner. To put the algorithms on more equal footing, we might allow the CG method to use a similar amount of storage and arithmetic operations to do some form of reorthogonalization. While there are many complicated schemes based on detailed analyses of the Lanczos algorithm in finite-precision arithmetic [57, 65], a simple scheme is to perform reorthogonalization only for the first $s\ell$ iterations and then to continue orthogonalizing against these vectors in subsequent iterations. In this case, the cost of reorthogonalization during the first $s\ell$ iterations will be $O(ds^2\ell^2)$ operations (the same as the cost to build the Nyström preconditioner), and subsequently orthogonalizing against these vectors will require $O(ds\ell)$ operations per iteration—the same as the cost to apply the Nyström preconditioner to a vector.

In Figure B.1b we display the behavior of the CG method with full reorthogonalization for the first 22 iterations and then orthogonalizing the subsequent Krylov basis vectors against

[8]We need some oversampling to capture the top eigenspace.

these first thirty vectors. Compared to Figure B.1a, we observe that CG now converges as rapidly as the Nyström PCG method.

B.5. The block-CG method. In Figure 7.1 we compare the performance of these methods when we do not use full reorthogonalization. In particular, we run the Nyström PCG method with $s = 3$ without any reorthogonalization. Building the preconditioner requires orthogonalizing $s\ell$ vectors. For the block-CG and CG methods, we use full reorthogonalization until we have a set of 3ℓ vectors and then continue to orthogonalize only against these vectors. For reference, we also display the convergence of the block-CG and the CG algorithm if no reorthogonalization is used at all. Unfortunately, far less is known about the behavior of the block-CG and block-Lanczos methods in finite-precision arithmetic than is known about their single-vector counterparts.

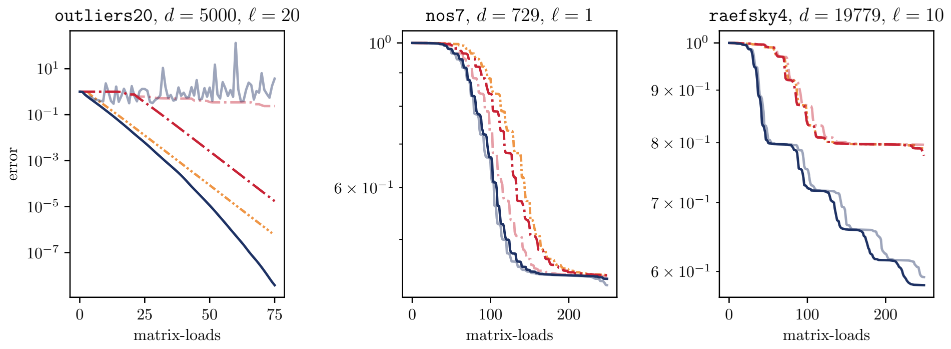


FIG. B.2. *The relative error $\|\mathbf{A}^{-1}\mathbf{b} - \mathsf{alg}\|_{\mathbf{A}}/\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ versus the matrix loads for the block-CG method (—) with reorthogonalization for 3 iterations, the CG method (–·–) with reorthogonalization for 3ℓ iterations, and the Nyström PCG method with $s = 3$ (–··–) without any reorthogonalization. The light curves show convergence of the block-CG and CG methods with no reorthogonalization. The test problems are the same as in Figure 7.1.*

Appendix C. Omitted proofs. In this section, we provide proofs that were omitted in the main text.

C.1. Proof of Proposition 4.1.

Proof of Proposition 4.1. Let $\mathbf{A}\langle\mathbf{K}\rangle$ have a rank-r eigendecomposition $\mathbf{UDU}^\mathsf{T}$, and define

$$\mathbf{E} = \mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle, \qquad \widehat{\mathbf{A}}_\mu = \mathbf{A}\langle\mathbf{K}\rangle + \mu\mathbf{I}.$$

Note that

$$\mathbf{P}_\mu^{-1} = (\theta + \mu)\mathbf{U}(\mathbf{D} + \mu\mathbf{I})^{-1}\mathbf{U}^\mathsf{T} + (\mathbf{I} - \mathbf{UU}^\mathsf{T}),$$

and observe that

$$\widehat{\mathbf{A}}_\mu = \mathbf{UDU}^\mathsf{T} + \mu\mathbf{I} = \mathbf{U}(\mathbf{D} + \mu\mathbf{I})\mathbf{U}^\mathsf{T} + \mu(\mathbf{I} - \mathbf{UU}^\mathsf{T}).$$

We begin with an upper bound for the eigenvalues of $\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}$. Using the definition of $\mathbf{E}$ and the triangle inequality,

$$\|\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}\| \leq \|\mathbf{P}_\mu^{-1/2}\widehat{\mathbf{A}}_\mu\mathbf{P}_\mu^{-1/2}\| + \|\mathbf{P}_\mu^{-1/2}\mathbf{E}\mathbf{P}_\mu^{-1/2}\|.$$

Using that $\mathbf{U}$ has orthonormal columns, we get

$$\mathbf{P}_\mu^{-1/2}\widehat{\mathbf{A}}_\mu\mathbf{P}_\mu^{-1/2} = (\theta+\mu)\mathbf{U}\mathbf{U}^\mathsf{T} + \mu(\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T})$$

and hence, since $\theta > 0$, $\|\mathbf{P}_\mu^{-1/2}\widehat{\mathbf{A}}_\mu\mathbf{P}_\mu^{-1/2}\| = \theta+\mu$. Again using that $\theta > 0$, we note that $\|\mathbf{P}^{-1/2}\| \leq 1$ and hence $\|\mathbf{P}_\mu^{-1/2}\mathbf{E}\mathbf{P}_\mu^{-1/2}\| \leq \|\mathbf{E}\|$. Therefore, we find that

$$\|\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}\| \leq \theta+\mu+\|\mathbf{E}\|. \tag{C.1}$$

We now derive a lower bound for the eigenvalues of $\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2}$. Noting that $(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})^{-1} = \mathbf{P}_\mu^{1/2}\mathbf{A}_\mu^{-1}\mathbf{P}_\mu^{1/2}$ is similar to $\mathbf{A}_\mu^{-1/2}\mathbf{P}_\mu\mathbf{A}_\mu^{-1/2}$, the definition of $\mathbf{P}_\mu$ and the triangle inequality implies

$$\begin{aligned}
\|(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})^{-1}\| &= \|\mathbf{A}_\mu^{-1/2}\mathbf{P}_\mu\mathbf{A}_\mu^{-1/2}\| \\
&= \left\|\mathbf{A}_\mu^{-1/2}\Big(\frac{1}{\theta+\mu}\mathbf{U}(\mathbf{D}+\mu\mathbf{I})\mathbf{U}^\mathsf{T} + (\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T})\Big)\mathbf{A}_\mu^{-1/2}\right\| \\
&\leq \frac{1}{\theta+\mu}\|\mathbf{A}_\mu^{-1/2}\mathbf{U}(\mathbf{D}+\mu\mathbf{I})\mathbf{U}^\mathsf{T}\mathbf{A}_\mu^{-1/2}\| + \|\mathbf{A}_\mu^{-1/2}(\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T})\mathbf{A}_\mu^{-1/2}\|.
\end{aligned}$$

Since $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ is obtained from a Nyström approximation to $\mathbf{A}$, $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T} \preceq \mathbf{A}$. Therefore, $\mathbf{U}(\mathbf{D}+\mu\mathbf{I})\mathbf{U}^\mathsf{T} \preceq \mathbf{A}+\mu\mathbf{I}$, and hence

$$\|\mathbf{A}_\mu^{-1/2}\mathbf{U}(\mathbf{D}+\mu\mathbf{I})\mathbf{U}^\mathsf{T}\mathbf{A}_\mu^{-1/2}\| \leq 1.$$

Likewise, since $\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T} \preceq \mathbf{I}$,

$$\begin{aligned}
\|\mathbf{A}_\mu^{-1/2}(\mathbf{I}-\mathbf{U}\mathbf{U}^\mathsf{T})\mathbf{A}_\mu^{-1/2}\| \leq \|\mathbf{A}_\mu^{-1}\| &= \frac{1}{\lambda_n+\mu}, \\
\|\mathbf{A}_\mu^{-1/2}\mathbf{P}_\mu\mathbf{A}_\mu^{-1/2}\| &\leq \frac{1}{\theta+\mu} + \frac{1}{\lambda_n+\mu}.
\end{aligned}$$

Therefore,

$$\|(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})^{-1}\| \leq \frac{1}{\theta+\mu} + \frac{1}{\lambda_n+\mu}. \tag{C.2}$$

Combining (C.1) and (C.2) gives the result. ◻

C.2. A bound for the elliptic integral.

LEMMA C.1. *For $m < 1$, define*

$$K(m) := \int_0^{\pi/2} (1-m\sin^2(z))^{-1/2}\mathrm{d}z.$$

Then, for all $x > 1$,

$$\sqrt{x}K(1-x) \leq \frac{5}{8}\log(16x).$$

Proof. The result [60, Theorem 1.3] asserts that if $m \in (0,1)$, then

$$K(m) \leq \log\left(\frac{4}{\sqrt{1-m}}\right)\left(1+\frac{1-m}{4}\right). \tag{C.3}$$

It is well known that $K(m)$ satisfies

$$K(m) = \frac{1}{\sqrt{1-m}} K\left(\frac{-m}{1-m}\right),$$

and hence,

$$K(1-x) \leq \frac{1}{\sqrt{x}} K\left(\frac{x-1}{x}\right).$$

Rearranging gives

$$\sqrt{x} K(1-x) \leq K\left(1 - \frac{1}{x}\right).$$

If $x > 1$, then $1 - 1/x \in (0,1)$, so we can apply (C.3) to obtain the bound

$$\sqrt{x} K(1-x) \leq \log\left(\frac{4}{\sqrt{1/x}}\right)\left(1 + \frac{1/x}{4}\right).$$

The result follows by noting that $(1 + 1/(4x))/2 \leq 5/8$. $\square$

REFERENCES

[1] O. Balabanov and L. Grigori, *Randomized block Gram-Schmidt process for the solution of linear systems and eigenvalue problems*, SIAM J. Sci. Comput., 47 (2025), pp. A553–A585.
[2] J. M. Bardsley, *MCMC-based image reconstruction with uncertainty quantification*, SIAM J. Sci. Comput., 34 (2012), pp. A1316–A1332.
[3] J. Besag, J. York, and A. Mollié, *Bayesian image restoration, with two applications in spatial statistics*, Ann. Inst. Statist. Math., 43 (1991), pp. 1–59.
[4] S. Birk and A. Frommer, *A deflated conjugate gradient method for multiple right hand sides and multiple shifts*, Numer. Algorithms, 67 (2014), pp. 507–529.
[5] D. Calvetti, L. Reichel, and A. Shuibi, *Tikhonov regularization of large symmetric problems*, Numer. Linear Algebra Appl., 12 (2005), pp. 127–139.
[6] E. Carson and I. Daužickaitė, *Single-pass Nyström approximation in mixed precision*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 1361–1391.
[7] E. Carson, K. Lund, M. Rozložník, and S. Thomas, *Block Gram-Schmidt algorithms and their stability properties*, Linear Algebra Appl., 638 (2022), pp. 150–195.
[8] T. Chen, *The Lanczos algorithm for matrix functions: a handbook for scientists*, Preprint on arXiv, 2024. https://arxiv.org/abs/2410.11090
[9] T. Chen, E. N. Epperly, R. A. Meyer, C. Musco, and A. Rao, *Does block size matter in randomized block Krylov low-rank approximation?*, in Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), K. G. Larsen and B. Saha, eds., SIAM, Philadelphia, 2025, pp. 1026–1046.
[10] T. Chen, A. Greenbaum, C. Musco, and C. Musco, *Error bounds for Lanczos-based matrix function approximation*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 787–811.
[11] T. Chen and E. Hallman, *Krylov-aware stochastic trace estimation*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1218–1244.
[12] S. Chenakkod, M. Dereziński, X. Dong, and M. Rudelson, *Optimal embedding dimension for sparse subspace embeddings*, in STOC'24—Proceedings of the 56th Annual ACM Symposium on Theory of Computing, B. Mohar, I. Shinkar, and R. O' Donnell, eds., ACM, New York, 2024, pp. 1106–1117.
[13] E. Chow and Y. Saad, *Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions*, SIAM J. Sci. Comput., 36 (2014), pp. A588–A608.
[14] K. L. Clarkson and D. P. Woodruff, *Low rank approximation and regression in input sparsity time*, in Proceedings of the 45th annual ACM symposium on Theory of Computing, STOC'13, ACM, New York, 2013, pp. 81–90.
[15] M. Dereziński, D. LeJeune, D. Needell, and E. Rebrova, *Fine-grained analysis and faster algorithms for iteratively solving linear systems*, J. Mach. Learn. Res., 26 (2025), Paper No. 144, 49 pages.

[16] M. DEREZIŃSKI, C. MUSCO, AND J. YANG, *Faster linear systems and matrix norm approximation via multi-level sketched preconditioning*, in Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadephia, 2025, pp. 1972–2004.
[17] M. DEREZIŃSKI, D. NEEDELL, E. REBROVA, AND J. YANG, *Randomized Kaczmarz methods with beyond-Krylov convergence*, SIAM J. Matrix Anal. Appl., 46 (2025), pp. 2558–2588.
[18] M. DEREZIŃSKI AND E. REBROVA, *Sharp analysis of sketch-and-project methods via a connection to randomized singular value decomposition*, SIAM J. Math. Data Sci., 6 (2024), pp. 127–153.
[19] M. DEREZIŃSKI AND J. YANG, *Solving dense linear systems faster than via preconditioning*, in STOC'24—Proceedings of the 56th Annual ACM Symposium on Theory of Computing, B. Mohar, I. Shinkar, and R. O' Donnell, eds., ACM, New York, 2024, pp. 1118–1129.
[20] V. L. DRUSKIN AND L. A. KNIZHNERMAN, *Error estimates in the simple Lanczos process in the calculation of functions of symmetric matrices and eigenvalues*, Comput. Math. Math. Phys., 31 (1991), p. 20–30.
[21] M. DÍAZ, E. N. EPPERLY, Z. FRANGELLA, J. A. TROPP, AND R. J. WEBBER, *Robust, randomized preconditioning for kernel ridge regression*, Preprint on arXiv, 2023.
https://arxiv.org/abs/2304.12465
[22] H. W. ENGL, M. HANKE, AND G. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.
[23] E. N. EPPERLY, J. A. TROPP, AND R. J. WEBBER, XTRACE*: making the most of every sample in stochastic trace estimation*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 1–23.
[24] Y. T. FENG, D. R. J. OWEN, AND D. PERIĆ, *A block conjugate gradient method applied to linear systems with multiple right-hand sides*, Comput. Methods Appl. Mech. Engrg., 127 (1995), pp. 203–215.
[25] Z. FRANGELLA, J. A. TROPP, AND M. UDELL, *Randomized Nyström preconditioning*, SIAM J. Matrix Anal. Appl., 44 (2023), p. 718–752.
[26] J. FRANK AND C. VUIK, *On the construction of deflation-based preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp. 442–462.
[27] A. FROMMER, S. GÜTTEL, AND M. SCHWEITZER, *Efficient and stable Arnoldi restarts for matrix functions based on quadrature*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 661–683.
[28] A. FROMMER AND P. MAASS, *Fast CG-based methods for Tikhonov-Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.
[29] A. FROMMER, G. RAMIREZ-HIDALGO, M. SCHWEITZER, AND M. TSOLAKIS, *Polynomial preconditioning for the action of the matrix square root and inverse square root*, Electron. Trans. Numer. Anal., 60 (2024), pp. 381–404.
https://etna.ricam.oeaw.ac.at/vol.60.2024/pp381-404.dir/pp381-404.pdf
[30] J. GARDNER, G. PLEISS, K. Q. WEINBERGER, D. BINDEL, AND A. G. WILSON, *Gpytorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration*, in Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Vol. 31, Curran Associates, Red Hook, 2018.
[31] G. H. GOLUB AND R. UNDERWOOD, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, Vol. 39 of Publication of the Mathematics Research Center, University of Wisconsin, Academic Press, New York-London, 1977, pp. 361–377.
[32] A. GREENBAUM, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.
[33] ———, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
[34] L. GRIGORI, S. MOUFAWAD, AND F. NATAF, *Enlarged Krylov subspace conjugate gradient methods for reducing communication*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 744–773.
[35] L. GRIGORI AND O. TISSOT, *Reducing the communication and computational costs of enlarged Krylov subspaces conjugate gradient*, Research Report RR-9023, Inria Paris, Paris, 2017.
[36] ———, *Scalable linear solvers based on enlarged Krylov subspaces with dynamic reduction of search directions*, SIAM J. Sci. Comput., 41 (2019), pp. C522–C547.
[37] M. H. GUTKNECHT, *Spectral deflation in Krylov solvers: a theory of coordinate space based methods*, Electron. Trans. Numer. Anal., 39 (2012), pp. 156–185.
https://etna.ricam.oeaw.ac.at/vol.39.2012/pp156-185.dir/pp156-185.pdf
[38] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
[39] M. HANKE, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman, Harlow, 1995.
[40] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[41] T. HONG, Z. XU, J. HU, AND J. A. FESSLER, *On adapting randomized Nyström preconditioners to accelerate variational image reconstruction*, Preprint on arXiv, 2024.
https://arxiv.org/abs/2411.08178v1
[42] M. KUSS AND C. E. RASMUSSEN, *Assessing approximate inference for binary Gaussian process classification*, J. Mach. Learn. Res., 6 (2005), pp. 1679–1704.
[43] H. LI, G. C. LINDERMAN, A. SZLAM, K. P. STANTON, Y. KLUGER, AND M. TYGERT, *Algorithm 971: an*

implementation of a randomized algorithm for principal component analysis, ACM Trans. Math. Software, 43 (2017), Paper No. 28, 14 pages.

[44] J. LOK AND E. REBROVA, *A subspace constrained randomized Kaczmarz method for structure or external knowledge exploitation*, Linear Algebra Appl., 698 (2024), pp. 220–260.

[45] P.-G. MARTINSSON AND M. O'NEIL, *Fast direct solvers*, Preprint on arXiv, 2025. https://arxiv.org/abs/2511.07773

[46] P.-G. MARTINSSON AND J. A. TROPP, *Randomized numerical linear algebra: foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572.

[47] G. MEURANT, *The block preconditioned conjugate gradient method on vector computers*, BIT, 24 (1984), pp. 623–633.

[48] ———, *The Lanczos and Conjugate Gradient Algorithms*, SIAM, Philadelphia, 2006.

[49] G. MEURANT AND P. TICHÝ, *Error norm estimates for the block conjugate gradient algorithms*, SIAM J. Matrix Anal. Appl., 46 (2025), pp. 2175–2196.

[50] R. MEYER, C. MUSCO, AND C. MUSCO, *On the unreasonable effectiveness of single vector Krylov methods for low-rank approximation*, in Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), D. P. Woodruff, ed., SIAM, Philadelphia, 2024, pp. 811–845.

[51] C. MUSCO AND C. MUSCO, *Randomized block Krylov methods for stronger and faster approximate singular value decomposition*, in Proceedings of the 29th International Conference on Neural Information Processing Systems, Vol. 1, C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett, eds., MIT Press, Cambridge, 2015, pp. 1396–1404.

[52] C. MUSCO, C. MUSCO, AND A. SIDFORD, *Stability of the Lanczos method for matrix function approximation*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, A. Czumaj, ed., SIAM, Philadelphia, 2018, pp. 1605–1624.

[53] Y. NAKATSUKASA AND J. A. TROPP, *Fast and accurate randomized algorithms for linear systems and eigenvalue problems*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 1183–1214.

[54] D. P. O'LEARY, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.

[55] C. C. PAIGE, *The computation of eigenvalues and eigenvectors of very large sparse matrices*, PhD. Thesis, University of London, London, 1971.

[56] ———, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[57] B. N. PARLETT AND D. S. SCOTT, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.

[58] B. A. PEARLMUTTER, *Fast exact multiplication by the Hessian*, Neural Comput., 6 (1994), pp. 147–160.

[59] G. PLEISS, M. JANKOWIAK, D. ERIKSSON, A. DAMLE, AND J. GARDNER, *Fast matrix square roots with applications to Gaussian processes and Bayesian optimization*, in Advances in Neural Information Processing Systems, Vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., Curran Associates, Red Hook, 2020, pp. 22268–22281.

[60] S.-L. QIU AND M. K. VAMANAMURTHY, *Sharp estimates for complete elliptic integrals*, SIAM J. Math. Anal., 27 (1996), pp. 823–834.

[61] V. ROKHLIN AND M. TYGERT, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 13212–13217.

[62] Y. SAAD, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[63] ———, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadephia, 2003.

[64] M. SCHWEITZER, *Near instance optimality of the Lanczos method for Stieltjes and related matrix functions*, SIAM J. Matrix Anal. Appl., 46 (2025), pp. 1846–1865.

[65] H. D. SIMON, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–142.

[66] W. R. THOMPSON, *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*, Biometrika, 25 (1933), pp. 285–294.

[67] P. TICHÝ, G. MEURANT, AND D. ŠIMONOVÁ, *Block CG algorithms revisited*, Numer. Algorithms, 100 (2025), pp. 1527–1553.

[68] J. A. TROPP AND R. J. WEBBER, *Randomized algorithms for low-rank matrix approximation: design, analysis, and applications*, Preprint on arXiv, 2023. https://arxiv.org/abs/2306.12418

[69] S. UBARU, J. CHEN, AND Y. SAAD, *Fast estimation of* $tr(f(A))$ *via stochastic Lanczos quadrature*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1075–1099.

[70] D. ŠIMONOVÁ AND P. TICHÝ, *On finite precision block Lanczos computations*, BIT, 65 (2025), Paper No. 47, 28 pages.

[71] Z. WANG AND S. JEGELKA, *Max-value entropy search for efficient bayesian optimization*, in Proceedings of the 34th International Conference on Machine Learning, Vol. 70, D. Precup and Y. W. Teh, eds., ICML'17, JMLR.org, 2017, pp. 3627–3635.

[72] Q. XU AND T. CHEN, *A posteriori error bounds for the block-Lanczos method for matrix function approximation*, Numer. Algorithms, 98 (2025), pp. 903–927.

[73] T. XU, D. CAI, H. HUANG, E. CHOW, AND Y. XI, *Preconditioned truncated single-sample estimators for scalable stochastic optimization*, Preprint on arXiv, 2025. https://arxiv.org/abs/2510.24587

[74] S. ZHAO, T. XU, H. HUANG, E. CHOW, AND Y. XI, *An adaptive factorized Nyström preconditioner for regularized kernel matrices*, SIAM J. Sci. Comput., 46 (2024), pp. A2351–A2376.

[75] J. ZIMMERLING, V. DRUSKIN, AND V. SIMONCINI, *Monotonicity, bounds and acceleration of block Gauss and Gauss–Radau quadrature for computing* $B^T\phi(A)B$, J. Sci. Comput., 103 (2025), Paper No. 5, 21 pages.