Preconditioning without a Preconditioner

Tyler Chen

JPMorganChase

Disclaimer

This presentation was prepared for informational purposes by the Global Technology Applied Research center of JPMorgan Chase & Co. This paper is not a merchandisable/sellable product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

Paper

Preconditioning without a preconditioner using randomized block Krylov subspace methods

Tyler Chen, Caroline Huber, Ethan Lin, Hajar Zaid

`https://arxiv.org/abs/2501.18717` (to appear in ETNA)

Context

I am broadly interested in understanding the following:

Why don’t we use block-KSMs for matrix functions?

Introduction

We are interested in solving the PSD linear system

$$\mathbf{A}_\mu \mathbf{x} = \mathbf{b}, \quad \mathbf{A}_\mu := \mathbf{A} + \mu \mathbf{I}.$$

This is relevant for:

- Solving PSD linear systems
- Ridge regression (Tikhonov regularization)
- Sampling Gaussian vectors

In the latter two examples, we might need the solution for many values μ efficiently.

Introduction

We are interested in solving the PSD linear system

$$\mathbf{A}_\mu \mathbf{x} = \mathbf{b}, \quad \mathbf{A}_\mu := \mathbf{A} + \mu \mathbf{I}.$$

This is relevant for:

- Solving PSD linear systems
- Ridge regression (Tikhonov regularization)
- Sampling Gaussian vectors

In the latter two examples, we might need the solution for many values μ efficiently.

Preconditioned Conjugate Gradient

CG produces an optimal solution over the Krylov subspace

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

If $\mathbf{A}_\mu$ is poorly conditioned, then CG might converge slowly. So it is common to work over a "preconditioned Krylov subspace":

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

PCG converges in

$$\sqrt{\kappa(\mathbf{P}_\mu^{-1/2}\mathbf{A}_\mu\mathbf{P}_\mu^{-1/2})}\log\left(\frac{1}{\varepsilon}\right) \text{ iterations.}$$

Need to choose preconditioner that is easy to construct and apply but also reduces condition number.

Deflation Preconditioner

Often, $\mathbf{A}_\mu$ is ill-conditioned because of r large eigenvalues: $\lambda_r \gg \lambda_{r+1} \approx \lambda_d$.

One can use the "deflation preconditioner"

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

where $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ is the eigendecomposition of the best rank-r approx to $\mathbf{A}$.

Top eigenvalues mapped to $\theta + \mu$, bottom eigenvalues untouched.

So if $\theta \in [\lambda_d + \mu, \lambda_{r+1} + \mu]$, we get convergence in

$$\sqrt{\kappa_r(\mathbf{A}_\mu)}\log\left(\frac{1}{\varepsilon}\right) \text{ iterations.}$$

Fast convergence!

Deflation Preconditioner

Often, $\mathbf{A}_\mu$ is ill-conditioned because of r large eigenvalues: $\lambda_r \gg \lambda_{r+1} \approx \lambda_d$.

One can use the "deflation preconditioner"

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

where $\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ is the eigendecomposition of the best rank-r approx to $\mathbf{A}$.

Top eigenvalues mapped to $\theta + \mu$, bottom eigenvalues untouched.

So if $\theta \in [\lambda_d + \mu, \lambda_{r+1} + \mu]$, we get convergence in

$$\sqrt{\kappa_r(\mathbf{A}_\mu)}\log\left(\frac{1}{\varepsilon}\right) \text{ iterations.}$$

Fast convergence!

Nyström low-rank approximation

Computing top eigenvectors is hard! However, RandNLA gives us very good approximate methods.

When $\mathbf{A}$ is positive semi-definite, a good choice is the Nyström approximation

$$\mathbf{A}\langle\mathbf{K}\rangle := (\mathbf{AK})(\mathbf{K}^T\mathbf{AK})^{-1}(\mathbf{K}^T\mathbf{A}).$$

We can build a preconditioner by taking $\mathbf{UDU}^T$ as the eigendecomposition of $\mathbf{A}\langle\mathbf{K}\rangle$.

Nyström low-rank approximation

Computing top eigenvectors is hard! However, RandNLA gives us very good approximate methods.

When $\mathbf{A}$ is positive semi-definite, a good choice is the Nyström approximation

$$\mathbf{A}\langle\mathbf{K}\rangle := (\mathbf{AK})(\mathbf{K}^\mathsf{T}\mathbf{AK})^{-1}(\mathbf{K}^\mathsf{T}\mathbf{A}).$$

We can build a preconditioner by taking $\mathbf{UDU}^\mathsf{T}$ as the eigendecomposition of $\mathbf{A}\langle\mathbf{K}\rangle$.

Nyström PCG[2]

Proposition.[1] Let $\mathbf{P}_\mu$ be the Nyström preconditioner corresponding to the Nyström approximation $\mathbf{A}\langle\mathbf{K}\rangle$ for any $\mathbf{K}$ and shift parameter $\theta \geq 0$. Then

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

If $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\| \approx \lambda_{\mathsf{r}+1}$, then Nyström PCG behaves similar to deflation!

Caveat: The space $\mathcal{K}_\mathsf{t}(\mathbf{A}_\mu, \mathbf{b}; \mathbf{P}_\mu)$ depends on μ! So we need to re-run PCG for each value of μ. (in contrast, $\mathcal{K}_\mathsf{t}(\mathbf{A}_\mu, \mathbf{b})$ does not depend on μ)

[1] Frangella, Tropp, and Udell 2023.

[2] Martinsson and Tropp 2020; Frangella, Tropp, and Udell 2023; Carson and Daužickaitė 2024; Díaz et al. 2023; Zhao et al. 2024; Hong et al. 2024; Dereziński, Musco, and Yang 2025.

Nyström PCG[2]

Proposition.[1] Let $\mathbf{P}_\mu$ be the Nyström preconditioner corresponding to the Nyström approximation $\mathbf{A}\langle\mathbf{K}\rangle$ for any $\mathbf{K}$ and shift parameter $\theta \geq 0$. Then

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

If $\|\mathbf{A} - \mathbf{A}\langle\mathbf{K}\rangle\| \approx \lambda_{\mathsf{r}+1}$, then Nyström PCG behaves similar to deflation!

Caveat: The space $\mathcal{K}_\mathsf{t}(\mathbf{A}_\mu, \mathbf{b}; \mathbf{P}_\mu)$ depends on μ! So we need to re-run PCG for each value of μ. (in contrast, $\mathcal{K}_\mathsf{t}(\mathbf{A}_\mu, \mathbf{b})$ does not depend on μ)

[1] Frangella, Tropp, and Udell 2023.

[2] Martinsson and Tropp 2020; Frangella, Tropp, and Udell 2023; Carson and Daužickaitė 2024; Díaz et al. 2023; Zhao et al. 2024; Hong et al. 2024; Dereziński, Musco, and Yang 2025.

Nyström PCG bounds

Theorem.[3] Suppose $\mathbf{K}$ is a Gaussian sketching matrix. If the sketching dimension is on the order of

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

then Nyström PCG converges in

$$\log\left(\frac{1}{\varepsilon}\right) \text{ iterations.}$$

Good: No dependence on the condition number!

To improve: How do we know the effective dimension? What if $\mu = 0$? What if there are multiple μ?

[3] Frangella, Tropp, and Udell 2023.

Nyström PCG bounds

Theorem.[3] Suppose $\mathbf{K}$ is a Gaussian sketching matrix. If the sketching dimension is on the order of

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

then Nyström PCG converges in

$$\log\left(\frac{1}{\varepsilon}\right) \text{ iterations.}$$

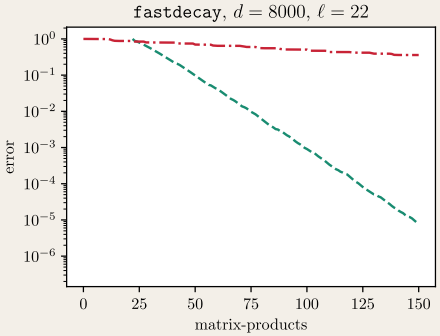
Good: No dependence on the condition number!

To improve: How do we know the effective dimension? What if $\mu = 0$? What if there are multiple μ?

[3] Frangella, Tropp, and Udell 2023.

Example

Test problem with a few outlying eigenvalues.



Legend: CG ( ) and Nyström PCG with s = 1 ( ).

Block Krylov Nyström

As a simple generalization, we consider block Krylov Nyström

$$\mathbf{A}\langle \mathbf{K}_s \rangle := (\mathbf{A}\mathbf{K}_s)(\mathbf{K}_s^T \mathbf{A}\mathbf{K}_s)^{-1}(\mathbf{K}_s^T \mathbf{A}).,\qquad \mathbf{K}_s := [\mathbf{G}\ \mathbf{A}\mathbf{G}\ \cdots\ \mathbf{A}^{s-1}\mathbf{G}].$$

Theorem.[4] Suppose $\mathbf{G} \in \mathbb{R}^{d\times(r+p)}$ is a random Gaussian matrix. Then, if $p \geq 2$,

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

This let's us prove a bound that says that if $s \approx \log(d)$, we get convergence in $\sqrt{\kappa_r(\mathbf{A}_\mu)}\log(1/\varepsilon)$ iterations.

To improve: Still don't know how to set block size $r + p$ or handle multiple μ.

[4] Tropp and Webber 2023.

Block Krylov Nyström

As a simple generalization, we consider block Krylov Nyström

$$\mathbf{A}\langle \mathbf{K}_s \rangle := (\mathbf{A}\mathbf{K}_s)(\mathbf{K}_s^T \mathbf{A}\mathbf{K}_s)^{-1}(\mathbf{K}_s^T \mathbf{A})., \qquad \mathbf{K}_s := [\mathbf{G}\ \mathbf{A}\mathbf{G}\ \cdots\ \mathbf{A}^{s-1}\mathbf{G}].$$

Theorem.[4] Suppose $\mathbf{G} \in \mathbb{R}^{d\times(r+p)}$ is a random Gaussian matrix. Then, if $p \geq 2$,

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

This let's us prove a bound that says that if $s \approx \log(d)$, we get convergence in $\sqrt{\kappa_r(\mathbf{A}_\mu)}\log(1/\varepsilon)$ iterations.

To improve: Still don't know how to set block size $r + p$ or handle multiple μ.

[4] Tropp and Webber 2023.

Block Krylov Methods

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

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

In at least some settings,[5] matrix-loads/iterations/matrix block-vector products are the dominant cost, in which case building a block Krylov subspace is nearly the same cost as building a single Krylov subspace.

[5] If it not realistic in your setting, please wait until the end of the talk to complain.

Block CG

Block CG is a generalization of CG that is optimal over the block Krylov subspace.

- If we care about $\mathbf{Ax} = \mathbf{b}^{(1)}$, block CG is no worse than CG (in terms of matrix-loads) because $\mathcal{K}_t(\mathbf{A}, \mathbf{b}^{(1)}) \subseteq \mathcal{K}_t(\mathbf{A}, \mathbf{B})$.

High-level question: When is block-CG better than CG?

Silly observation[6]

Theorem. Suppose $\mathbf{P}_\mu = (\mathbf{I} + \mathbf{X})^{-1}$, where $\mathsf{range}(\mathbf{X}) \subseteq \mathcal{K}_{s+1}(\mathbf{A}_\mu, \mathbf{G})$. Then,

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

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

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

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

[6] Chen, Huber, et al. 2026.

Silly main theorem[7]

Theorem. Fix any matrix $\mathbf{G} \in \mathbb{R}^{d \times m}$ and let $\mathbf{P}_\mu = (\mathbf{I} + \mathbf{X})^{-1}$ be any preconditioner where $\mathrm{range}(\mathbf{X}) \subseteq \mathcal{K}_{s+1}(\mathbf{A}, \mathbf{G})$. Define the augmented starting block $\mathbf{B} = [\mathbf{b}\ \mathbf{G}]$. 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 $\mathbf{P}_\mu$ in that

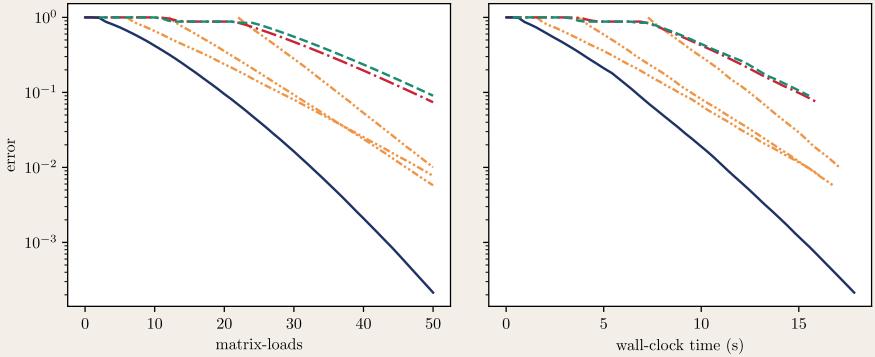
$$\|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathrm{bcg}_t^{(1)}(\mu)\|_{\mathbf{A}_\mu} \leq \|\mathbf{A}_\mu^{-1}\mathbf{b} - \mathrm{pcg}_{t-s}(\mu)\|_{\mathbf{A}_\mu}.$$

In short: augmented block-CG automatically performs no worse than Nyström PCG (with the best choice of s and θ) after the same number of matrix-loads.

- Our new bounds for Nyström PCG immediately give newbounds for augmented block CG (where $\mathbf{G}$ is Gaussian).

[7] Chen, Huber, et al. 2026.

Example



Legend: ours ( ), standard CG ( ), Nyström PCG from Frangella, Tropp, and Udell 2023 ( ), and generalizations Nyström PCG using larger s ( ).

Are we measuring costs correctly?

Earlier, I asked you not to complain about my access model (counting block-matvecs).[8]

The problem is, the story is actually kind of subtle, and there is no clear winner...

[8] The referees definitly did complain about this :/

What are the costs?

For Nyström PCG:

- Nyström approximation with starting block size ℓ requires s matrix block-vector products with **A** (each requires ℓ matvecs)
- Each iteration of PCG requires 1 matvec with **A**
- Unclear how to choose s, ℓ.
- We require a new preconditioner for every value of μ.

For block CG:

- Each iteration of BCG requires 1 block-vector products with **A**
- Unclear how to choose ℓ.
- Very cheap to solve for multiple values of μ.

Exact arithmetic CG

In exact arithmetic, vanilla CG already satisfies bounds in terms of $\kappa_r(\mathbf{A}_\mu)$.

Theorem. For any $r \geq 0$ the CG iterate satisfies

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

Proof. Recall the CG error can be bounded in terms of

$$\min_{\substack{\deg(p)=t \\ p(0)=1}} \left(\max_{x\in\mathrm{spec}(\mathbf{A})} |p(x)|\right).$$

To get the usual bound, we relax to $[\lambda_d, \lambda_1]$ and use use a shifted and scaled Chebyshev polynonial.

Instead, relax to $[\lambda_d, \lambda_{r+1}] \cup \{\lambda_r, \ldots, \lambda_1\}$, and place explicit zeros at $\lambda_r, \ldots, \lambda_1$.

Finite precision arithmetic

We can use the previous bound to prove a bound in terms of the effective dimension

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

Theorem.[9] The CG iterate satisfies

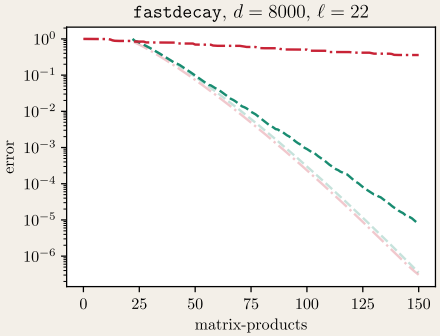
$$\frac{\|\mathbf{A}_{\mu}^{-1}\mathbf{b} - cg_t\|_{\mathbf{A}_{\mu}}}{\|\mathbf{A}_{\mu}^{-1}\mathbf{b} - cg_0\|_{\mathbf{A}_{\mu}}} \leq 2\exp\left(-\sqrt{2}(t - 2d_{\text{eff}}(\mu))\right).$$

Takeaway: The value of Nyström PCG over CG (in terms of matvecs) is predicated on being in finite preicsion arithmetic.

[9] Has anyone seen this bound before? It is immediatle from the fact that $r > 2d_{\text{eff}}(\mu) \implies \lambda_{r+1} \leq \mu$.

Example

Test problem with a few outlying eigenvalues (light=full reorth).



Legend: CG ( ) and Nyström PCG with s = 1 ( )

What if CG gets a little orthogonalization?

Full reorthogonalization might be too expensive.

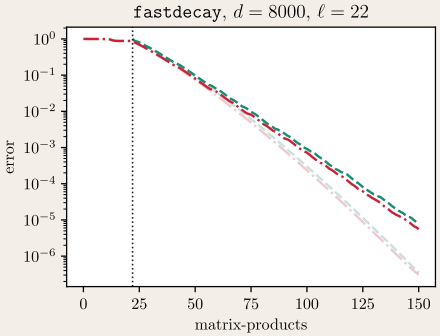
However:

- to build the Nyström preconditioner, we needed to orthogonalize ℓ vectors
- to apply the Nyström preconditioner, we need to compute ℓ inner products

If we run CG with orthogonalization for ℓ steps, and then only orthogonalize against these vectors, the compuational profile is very similar to Nyström PCG.

Example

Let CG orthogonalize for the first ℓ steps.



Legend: CG ( ) and Nyström PCG with s $= 1$ ( ).

Discussion

- CG doesn't need to set ℓ in advance and get sets similar performace. So does this mean Nyström PCG is pointless?
  - Not necessarily. Because the ℓ matvecs used in the sketch can be done in parallel (whereas CG does them sequentially).
- But if this is cheaper than doign ℓ seperate matvecs, then you need to start thinking about augmented block-CG.
  - And the same ideas don't require you to build the entire block Krylov subspace, you can try to do some kind of deflation.

My take: I don't really think augmented block-CG in the form described here it is a good idea in most cases. But the overall idea of just building Krylov subspaces instead of preconditioners is promising.

References I

Carson, Erin and Ieva Daužickaitė (July 2024). “Single-Pass Nyström Approximation in Mixed Precision”. In: SIAM Journal on Matrix Analysis and Applications 45.3, pp. 1361–1391. ISSN: 1095-7162. DOI: 10.1137/22m154079x.

Chen, Tyler, Ethan N. Epperly, et al. (2025). “Does block size matter in randomized block Krylov low-rank approximation?” In: Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). arXiv: 2508.06486 [cs.DS].

Chen, Tyler, Caroline Huber, et al. (2026). “Preconditioning without a preconditioner: faster ridge-regression and Gaussian sampling with randomized block Krylov subspace methods”. In: ETNA (to appear). arXiv: 2501.18717 [math.NA].

Dereziński, Michał, Christopher Musco, and Jiaming Yang (Jan. 2025). Faster Linear Systems and Matrix Norm Approximation via Multi-level Sketched Preconditioning. DOI: 10.1137/1.9781611978322.62.

Díaz, Mateo et al. (2023). Robust, randomized preconditioning for kernel ridge regression. arXiv: 2304.12465 [math.NA].

References II

Frangella, Zachary, Joel A. Tropp, and Madeleine Udell (May 2023). “Randomized Nyström Preconditioning”. In: SIAM Journal on Matrix Analysis and Applications 44.2, pp. 718–752. ISSN: 1095-7162. DOI: 10.1137/21m1466244.

Gardner, Jacob et al. (2018). “GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration”. In: Advances in Neural Information Processing Systems. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc.

Hong, Tao et al. (2024). On Adapting Randomized Nyström Preconditioners to Accelerate Variational Image Reconstruction. arXiv: 2411.08178 [eess.IV].

Martinsson, Per-Gunnar and Joel A. Tropp (May 2020). “Randomized numerical linear algebra: Foundations and algorithms”. In: Acta Numerica 29, pp. 403–572. ISSN: 1474-0508. DOI: 10.1017/s0962492920000021.

Tropp, Joel A. and Robert J. Webber (2023). Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications. arXiv: 2306.12418 [math.NA].

References III

Xu, Tianshi et al. (2025). Preconditioned Truncated Single-Sample Estimators for Scalable Stochastic Optimization. arXiv: 2510.24587 [math.NA].

Zhao, Shifan et al. (July 2024). “An Adaptive Factorized Nyström Preconditioner for Regularized Kernel Matrices”. In: SIAM Journal on Scientific Computing 46.4, A2351–A2376. ISSN: 1095-7197. DOI: 10.1137/23m1565139.

Miscellaneous

What about the block-size?

So far: set $\ell = O(r)$ and run for at least $s = \log(d)$ iterations.

But what if we don't know r ahead of time!?

- Since $\mathbf{A}\langle \mathbf{K}_s \rangle$ has rank as large as ℓs, we might hope that things are okay as long as $\ell s = O(r)$.

Theorem.[10] Let $\in \mathbb{R}^{d \times \ell}$ be a Gaussian matrix. Then, for some

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

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

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

Then easily extend this bound to Nyström.

[10] Chen, Epperly, et al. 2025.

Sampling Gaussians

Some applications ask to sample from $\mathcal{N}(\mathbf{m}, \mathbf{A})$. This can be done by

$$\mathbf{m} + \mathbf{A}^{1/2}\mathbf{b} \sim \mathcal{N}(\mathbf{m}, \mathbf{A}), \qquad \mathbf{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

There is a nice identity:

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

At each point z, we need to solve a shifted linear system $(\mathbf{A} + z^2\mathbf{I})\mathbf{x} = \mathbf{b}$.

Lanczos method for matrix functions

Let $\mathbf{Q} \in \mathbb{R}^{d \times t}$ and $\mathbf{T} \in \mathbb{R}^{t \times t}$ be the output of Lanczos run on $(\mathbf{A}, \mathbf{b})$ for t steps. Then the CG approximation to $(\mathbf{A} + \mu \mathbf{I})\mathbf{x} = \mathbf{b}$ is

$$\mathrm{cg}_t(\mu) := \|\mathbf{b}\| \mathbf{Q} (\mathbf{T} + \mu \mathbf{I})^{-1} \mathbf{e}_1.$$

By approximating each term of the integrand of our identity we get an formula

$$\mathbf{A}^{1/2}\mathbf{b} \approx \|\mathbf{b}\| \frac{2}{\pi} \int_0^\infty \mathbf{A}\mathbf{Q}(\mathbf{T} + z^2\mathbf{I})^{-1}\mathbf{e}_1 \, dz = \|\mathbf{b}\| \mathbf{A}\mathbf{Q}\mathbf{T}^{-1/2}\mathbf{e}_1,$$

which we can compute efficiently.

Error for Lanczos for matrix functions

To analyze the error:

$$\begin{aligned}\|\mathbf{A}^{1/2}\mathbf{b} - \|\mathbf{b}\|\mathbf{A}\mathbf{Q}\mathbf{T}^{-1/2}\mathbf{e}_1\| &= \left\| \frac{2}{\pi}\int_0^\infty \mathbf{A}(\mathbf{A}+z^2\mathbf{I})^{-1}\mathbf{b}\,dz - \frac{2}{\pi}\int_0^\infty \mathbf{A}\mathrm{cg}_t(z^2)\,dz \right\| \\ &\leq \frac{2}{\pi}\int_0^\infty \|\mathbf{A}(\mathbf{A}+z^2\mathbf{I})^{-1}\mathbf{b} - \mathrm{cg}_t(z^2)\|\,dz.\end{aligned}$$

Using augmented block-CG to analyze Block Lanczos-FA

Suppose we want to sample m Gaussians. We need to compute $\mathbf{A}^{1/2}\mathbf{b}^{(1)}, \ldots, \mathbf{A}^{1/2}\mathbf{A}^{1/2}\mathbf{b}^{(m)}$, where $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m)}$ be standard Gaussian vectors.

On the last slide, we saw how to reduce this to solving

$$(\mathbf{A} + z^2\mathbf{I})\mathbf{x} = \mathbf{b}^{(1)}.$$

Note that $[\mathbf{b}^{(2)}, \ldots, \mathbf{b}^{(m)}]$ is a Gaussian matrix independent of $\mathbf{b}^{(1)}$! So we can uase augmented BCG with these vectors as the Gaussian sketch.

Same argument for $\mathbf{b}^{(2)}, \ldots, \mathbf{b}^{(m)}$.

But the block Krylov subspace is the same for all of them, so we only need to build it once!

Convergence gurantees

High-level question: when do block KSMs for matrix functions outperform running a single-vector algorithm on each vector?

Lanczos-FA:

$$m\sqrt{\kappa}\log(1/\varepsilon) \text{ matrix-vector products.}$$

Block Lanczos-FA:

$$m\left(\log(d) + \sqrt{\kappa_{r+1}}\log\left(\log(\kappa)/\varepsilon\right)\right) \text{ matrix-vector products,}$$

where $r = O(m/\log(m))$.

Use in Gaussian Process regression

In certain applications, we need to compute $\mathbf{A}^{-1}\mathbf{b}$ and $\mathrm{tr}(\log(\mathbf{A}))$.

If we use Girard–Hutchinson estimator for the trace and Lanczos for the matrix function, then we end up a block Krylov subspace on random vectors. And people have been appending on $\mathbf{b}$ and then using this augmented block Krylov subspace.[11].

[11] Gardner et al. 2018; Xu et al. 2025.