Everything is Vecchia: Unifying low-rank and sparse inverse Cholesky approximations

Robert (Rob) J. Webber
Assistant Professor, Mathematics

February 5, 2026

Table of Contents

Framing the project

Past work: Randomized numerical linear algebra (RNLA) has generated low-rank approximations with strong theoretical guarantees:

- Randomized SVD
- Generalized Nystrom
- Randomized block Krylov.

Problem: What if the matrix is not approximately low-rank? How do we solve linear algebra problems then?

Goal: We need a flexible class of matrix approximations that includes low-rank approximation and admits strong guarantees.

Partial progress: The Vecchia approximation is a solid option. It includes low-rank approximations and other *mysterious* structures. We should understand it better.

Outlook: The Vecchia approximation will not solve positive-semidefinite approximation, but it will take us closer.

[Joint work with Eagan Kaminetz, arXiv paper coming out next week.]

Cholesky decomposition

Let us focus on positive-semidefinite matrices $\boldsymbol{A} \in \mathbb{C}^{n \times n}$.

They can be represented by a Cholesky or inverse Cholesky decomposition

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^*\boldsymbol{P}^* \quad \text{or} \quad \boldsymbol{A} = \boldsymbol{P}\boldsymbol{C}^{-1}\boldsymbol{D}\boldsymbol{C}^{-*}\boldsymbol{P}^*.$$

Here, $\boldsymbol{P} \in \{0,1\}^{n \times n}$ is a permutation matrix, $\boldsymbol{L}$ is lower triangular with ones on the diagonal, and $\boldsymbol{D}$ is diagonal with nonnegative entries.

These decompositions exist for every positive-semidefinite matrix $\boldsymbol{A}$ and permutation matrix $\boldsymbol{P}$, they are equivalent using $\boldsymbol{L} = \boldsymbol{C}^{-1}$.

Given such a factorization, we can efficiently solve linear systems, compute determinants, etc. in $\mathcal{O}(n^2)$ or $\mathcal{O}(n)$ time.

Cholesky decomposition

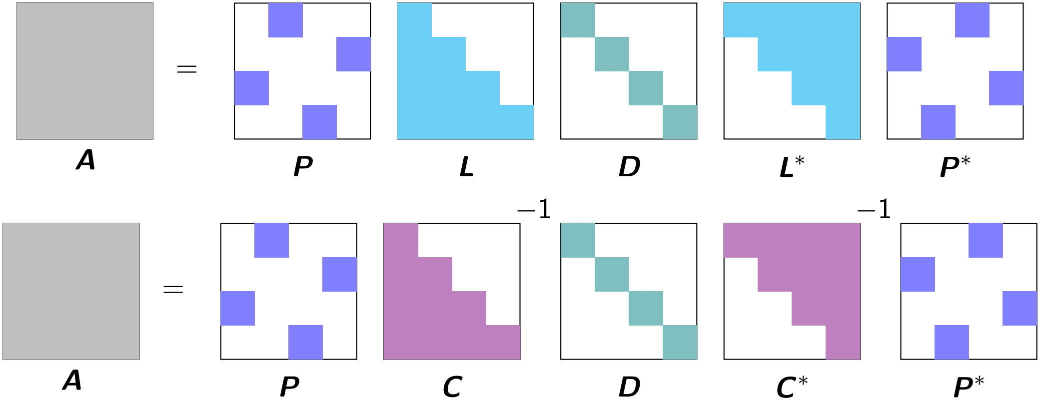


Figure: Cholesky and inverse Cholesky decompositions of a dense matrix $\boldsymbol{A}$. Factors $\boldsymbol{P}, \boldsymbol{L}, \boldsymbol{D}$ or $\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{D}$ are stored, and inverses $\boldsymbol{C}^{-1}, \boldsymbol{C}^{-*}$ are accessed implicitly. Filled boxes show entries that are allowed to be nonzero.

Sparse Cholesky approximation

Recent work uses *sparse* Cholesky or inverse Cholesky approximations

$$\hat{\boldsymbol{A}} = \boldsymbol{P}\hat{\boldsymbol{L}}\hat{\boldsymbol{D}}\hat{\boldsymbol{L}}^{*}\boldsymbol{P}^{*} \quad \text{or} \quad \hat{\boldsymbol{A}} = \boldsymbol{P}\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{D}}\hat{\boldsymbol{C}}^{-*}\boldsymbol{P}^{*}.$$

Here, $\hat{\boldsymbol{L}}$ or $\hat{\boldsymbol{C}}$ is very sparse (but $\hat{\boldsymbol{L}}^{-1}$ or $\hat{\boldsymbol{C}}^{-1}$ may not be)

- "randomized Cholesky" (Kyng, Sachdeva),
- "sparse Cholesky" (Schäfer, Katzfuss, Owhadi),
- "randomly pivoted Cholesky" (Chen, Epperly, Tropp, Webber)

– The sparsity pattern $\mathsf{S}_i \subseteq \{1, \ldots, i-1\}$ lists the nonzero off-diagonal entries in row i of $\hat{\boldsymbol{L}}$ or $\hat{\boldsymbol{C}}$.

– If $|\mathsf{S}_i| \leq s$ for $i = 1, \ldots, n$, we can solve linear systems in $\mathcal{O}(ns)$ operations.

– Often we can generate $\hat{\boldsymbol{A}}$ in $\mathcal{O}(s^2 n)$ or $\mathcal{O}(s^3 n)$ operations, cheaper than the cost of looking at each entry once.

Sparse Cholesky approximation

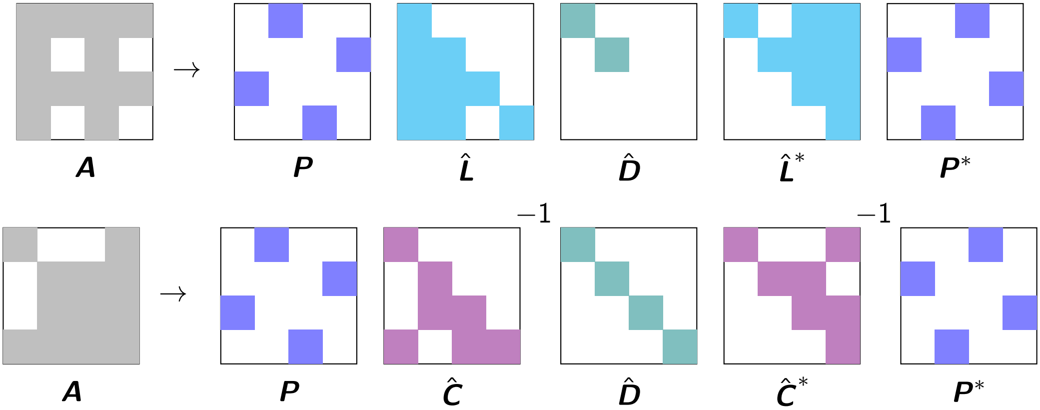


Figure: Sparse Cholesky and inverse Cholesky approximations of a dense matrix A. Filled boxes show entries that are allowed to be nonzero.

Partial pivoted Cholesky

Partial pivoted Cholesky generates a rank-r sparse Cholesky approximation that matches user-selected columns, indexed by $u_1, \ldots, u_r$.

Forming this approximation requires $\mathcal{O}(rn)$ entry look-ups and $\mathcal{O}(r^2 n)$ extra processing.

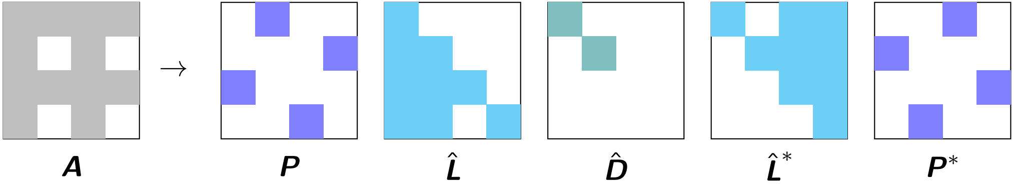
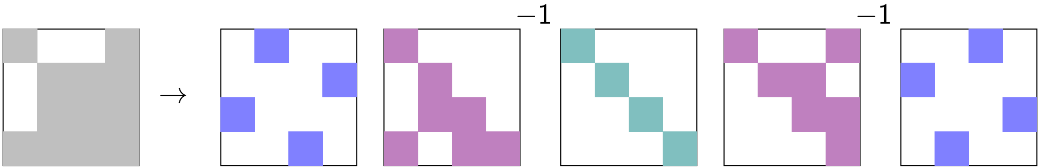


Figure: Partial pivoted Cholesky accesses gray-colored entries of $\boldsymbol{A}$. The rank is $r = 2$. Columns $u_1 = 3$ and $u_2 = 1$ are perfectly replicated.

Vecchia

Vecchia approximation (named after Aldo Vecchia) generates a sparse inverse Cholesky approximation with any user-selected permutation $\boldsymbol{P}$ and sparsity pattern $\{\mathrm{S}_i\}_{i=1}^{r}$.

Traditionally requires $\mathcal{O}(s^2 n)$ entry look-ups and $\mathcal{O}(s^3 n)$ extra processing, where s is an upper bound on the cardinality, $|\mathrm{S}_i| \leq s$.



$$\boldsymbol{A} \rightarrow \boldsymbol{P} \, \hat{\boldsymbol{C}}^{-1} \, \hat{\boldsymbol{D}} \, \hat{\boldsymbol{C}}^{*-1} \, \boldsymbol{P}^*$$

Figure: Vecchia approximation accesses gray entries of $\boldsymbol{A}$. Pivots are $u_1 = 3$, $u_2 = 1$, $u_3 = 4$, $u_4 = 2$. Sparsity pattern is $\mathrm{S}_2 = \emptyset$, $\mathrm{S}_3 = \{2\}$, $\mathrm{S}_4 = \{1, 3\}$.

Formula for the Vecchia approximation

Vecchia approximation uses $\hat{\boldsymbol{C}}(i, \mathrm{S}_i)$ and $\hat{\boldsymbol{D}}(i, i)$ that solve

$$\begin{bmatrix} \hat{\boldsymbol{C}}(i, \mathrm{S}_i) & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{A}(\mathrm{S}_i, \mathrm{S}_i) & \boldsymbol{A}(\mathrm{S}_i, i) \\ \boldsymbol{A}(i, \mathrm{S}_i) & \boldsymbol{A}(i, i) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \hat{\boldsymbol{D}}(i, i) \end{bmatrix}, \qquad \text{for } i = 1, \ldots, n.$$

Partial Cholesky + Vecchia

Traditionally, partial Cholesky and Vecchia are used for different tasks:

- Partial Cholesky exposes **low-rank structure**.
- Vecchia exposes **sparse structure** in the inverse Cholesky factor.

Edmond Chow, Yuanzhe Xi, and colleagues (2024 SISC, 2025 SIMAX) combined the approximations, by forming a partial Cholesky approximation *first* and forming a residual Vecchia approximation *second*.

The Chow & Xi papers were empirical, with promising numerical experiments. I wanted to find out more.

Partial Cholesky + Vecchia

Theorem (Partial Cholesky + Vecchia = Vecchia)

Given a target positive-semidefinite matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$*, consider the following two-part approximation.*

1. *Generate a partial Cholesky approximation of* $\boldsymbol{A}$ *with permutation* $\boldsymbol{P}$ *and approximation rank* r*. Call it* $\hat{\boldsymbol{A}}_{\mathrm{part}}$*.*
2. *Generate a Vecchia approximation of the residual* $\boldsymbol{R} = \boldsymbol{A} - \hat{\boldsymbol{A}}_{\mathrm{part}}$ *with permutation* $\boldsymbol{P}$ *and sparsity pattern* $\{\mathsf{Q}_i\}_{i=1}^{n}$*. Call it* $\hat{\boldsymbol{A}}_{\mathrm{res}}$*.*

Then $\hat{\boldsymbol{A}}_{\mathrm{part}} + \hat{\boldsymbol{A}}_{\mathrm{res}}$ *can be rewritten as a Vecchia approximation of* $\boldsymbol{A}$ *with permutation* $\boldsymbol{P}$ *and an augmented sparsity pattern* $\mathsf{S}_i = (\{1, \ldots, r\} \cup \mathsf{Q}_i) \cap \{1, \ldots, i-1\}$*.*

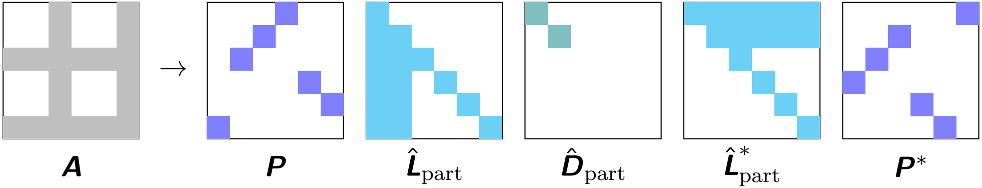
Proof.

Proof in pictures (next three slides). □

Partial Cholesky + Vecchia

Partial Cholesky accesses $\boldsymbol{A}$ and generates

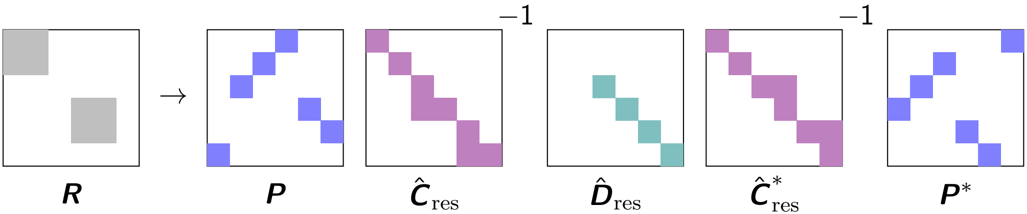
$$\hat{\boldsymbol{A}}_{\text{part}} = \boldsymbol{P}\hat{\boldsymbol{L}}_{\text{part}}\hat{\boldsymbol{D}}_{\text{part}}\hat{\boldsymbol{L}}^{*}_{\text{part}}\boldsymbol{P}^{*}.$$



Partial Cholesky + Vecchia

Vecchia uses $\boldsymbol{R} = \boldsymbol{A} - \hat{\boldsymbol{A}}_{\mathrm{part}}$ to generate

$$\hat{\boldsymbol{A}}_{\mathrm{res}} = \boldsymbol{P}\hat{\boldsymbol{C}}_{\mathrm{res}}^{-1}\hat{\boldsymbol{D}}_{\mathrm{res}}\hat{\boldsymbol{C}}_{\mathrm{res}}^{-*}\boldsymbol{P}^*.$$

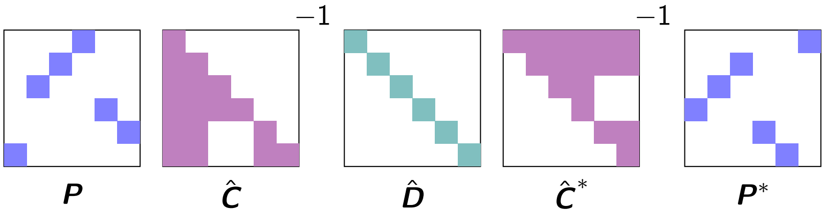


Partial Cholesky + Vecchia

Partial Cholesky + Vecchia can be rewritten

$$\begin{aligned}\hat{\boldsymbol{A}} &= \hat{\boldsymbol{A}}_{\mathrm{part}} + \hat{\boldsymbol{A}}_{\mathrm{res}} \\ &= \boldsymbol{P}\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{D}}\hat{\boldsymbol{C}}^{-*}\boldsymbol{P}^{*}.\end{aligned}$$

Hybrid →



This is a sparse inverse Cholesky decomposition with the right permutation and sparsity pattern. The rest of the proof just checks the Vecchia linear systems. ∎

Why is this important?

Partial Cholesky + Vecchia is often **computationally efficient**.

+ It reduces Vecchia's cost from $\mathcal{O}(s^2 n)$ entry look-ups and $\mathcal{O}(s^3 n)$ operations to $\mathcal{O}(sn)$ look-ups and $\mathcal{O}(s^2 n)$ operations for a special sparsity pattern with $|\mathsf{S}_i| \leq s$.

Partial Cholesky + Vecchia is often **accurate**.

+ In the simplest case, we apply partial Cholesky to $\boldsymbol{A}$ and then apply Vecchia with the minimal sparsity pattern $\mathsf{S}_i = \emptyset$. This leads to partial Cholesky + diagonal

$$\hat{\boldsymbol{A}} = \hat{\boldsymbol{A}}_{\text{part}} + \operatorname{diag}(\boldsymbol{A} - \hat{\boldsymbol{A}}_{\text{part}}).$$

+ Alternatively, we can apply partial Cholesky to $\boldsymbol{A}$ and then apply Vecchia with a small, carefully chosen sparsity pattern.

Experimental setup

We downloaded 27 machine learning data sets with 4–784 predictors and 1 quantitative response (OpenML, LibSVM, ...). We subsampled $n = 20,000$ data points.

We standardized the predictors and formed the $n \times n$ kernel matrix with entries

$$\boldsymbol{A}(i,j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2d}\right) + \mu\,\delta(i,j), \qquad \text{where } \mu \in \left\{10^{-3}, 10^{-6}, 10^{-10}\right\}.$$

We formed various matrix approximations $\hat{\boldsymbol{A}}$ and tested performance as follows.

1. Use $\hat{\boldsymbol{A}}$ as a preconditioner to solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{b}$ is the vector of labels.
2. Use $\log\det(\hat{\boldsymbol{A}}))$ as a direct estimator of $\log(\det(\boldsymbol{A}))$.
3. Use $\hat{\boldsymbol{A}}$ as a preconditioner for stochastic log determinant estimation.

Experimental results

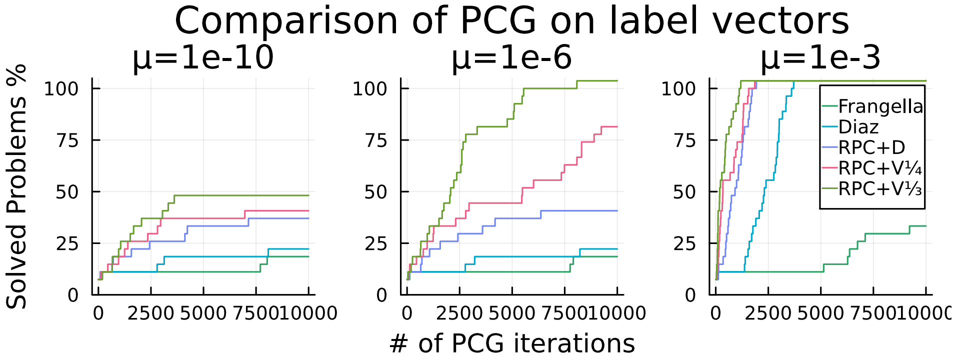


Figure: We ran randomly pivoted Cholesky (RPC) with rank $r = n^{1/2} = 141$ and used either (1) Frangella, Tropp, & Udell (2023); (2) Díaz, Epperly, Frangella, Tropp, & Webber (2023); (3) partial Cholesky + diagonal; (4) partial Cholesky + Vecchia with $q = n^{1/4} = 11$ nonzeros in the residual sparsity pattern; or (5) partial Cholesky + Vecchia with $q = n^{1/3} = 27$ nonzeros.

Experimental results

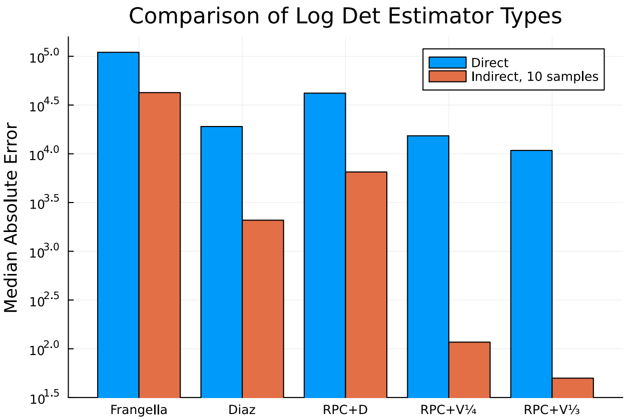


Figure: We set $\mu = 10^{-3}$ and estimated $\log(\det(\boldsymbol{A}))$ using (a) the direct estimator $\log(\det(\hat{\boldsymbol{A}}))$ or (b) the stochastic log determinant estimator with $\hat{\boldsymbol{A}}$ as a preconditioner.

Takeaways

Comparisons:

- For near-singular matrices, partial Cholesky + Vecchia provides the best approximation that's based on a partial Cholesky approximation.
- Partial Cholesky plus diagonal is okay but the Vecchia component helps a lot, especially with determinant estimation.

Outlook:

- All approximations fail, but partial Cholesky + Vecchia fails more slowly as the eigenvalue lower bound $\mu \downarrow 0$.

Vecchia optimality

Axelsson & Kaporin (1994, 2000) developed Vecchia optimality theory, and we generalized it to positive-semidefinite matrices.

Definition (Kaporin condition number)

For any positive-semidefinite matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ and any positive-semidefinite approximation $\hat{\boldsymbol{A}} \in \mathbb{C}^{n \times n}$, the Kaporin condition number is

$$\kappa_{\text{Kap}} = \frac{\left(\frac{1}{r}\operatorname{tr}(\boldsymbol{A}\hat{\boldsymbol{A}}^{+})\right)^{r}}{\operatorname{vol}(\boldsymbol{A}\hat{\boldsymbol{A}}^{+})}, \qquad \text{where } r = \operatorname{rank}(\boldsymbol{A}),$$

if $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ share the same range. The Kaporin condition number is $\kappa_{\text{Kap}} = \infty$ if $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ have different ranges.

κ_{Kap} is the average positive eigenvalue raised to the rank($\boldsymbol{A}$) power, divided by the product of the positive eigenvalues. By the AM-GM inequality, $\kappa_{\text{Kap}} \geq 1$.

Vecchia optimality

Theorem (Vecchia optimality)

For any positive-semidefinite matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$*, the Vecchia approximation* $\hat{\boldsymbol{A}} = \boldsymbol{P}\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{D}}\hat{\boldsymbol{C}}^{-*}\boldsymbol{P}^*$ *is the inverse Cholesky approximation with permutation* $\boldsymbol{P}$ *and sparsity pattern* $\{\mathsf{S}_i\}_{i=1}^n$ *that achieves the smallest possible Kaporin condition number. When* $\hat{\boldsymbol{A}}$ *and* $\boldsymbol{A}$ *have the same range, the Kaporin condition number is*

$$\kappa_{\mathrm{Kap}} = \prod_{d_{\tilde{\boldsymbol{A}}}(\boldsymbol{e}_i, \mathrm{span}\{\boldsymbol{e}_j\}_{j<i})>0} \frac{d_{\tilde{\boldsymbol{A}}}\left(\boldsymbol{e}_i, \mathrm{span}\{\boldsymbol{e}_j\}_{j\in\mathsf{S}_i}\right)^2}{d_{\tilde{\boldsymbol{A}}}\left(\boldsymbol{e}_i, \mathrm{span}\{\boldsymbol{e}_j\}_{j<i}\right)^2},$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{P}^*\boldsymbol{A}\boldsymbol{P}$ *is the permuted* $\boldsymbol{A}$ *matrix.*

Implications of Vecchia optimality

κ_{Kap} is important since it controls the error in linear algebra calculations.

Method	Error bound
Linear system, direct solver	$\frac{\lVert \hat{\boldsymbol{x}} - \boldsymbol{x}_\star \rVert_{\boldsymbol{A}}^2}{\lVert \boldsymbol{x}_0 - \boldsymbol{x}_\star \rVert_{\boldsymbol{A}}^2} \leq 2\,\mathrm{rank}(\boldsymbol{A}) \log(\kappa_{\mathrm{Kap}})$
Linear system, iterative solver	$\frac{\lVert \boldsymbol{x}_t - \boldsymbol{x}_\star \rVert_{\boldsymbol{A}}^2}{\lVert \boldsymbol{x}_0 - \boldsymbol{x}_\star \rVert_{\boldsymbol{A}}^2} \leq \left[\frac{3\log(\kappa_{\mathrm{Kap}})}{t}\right]^t$
Determinant, direct solver	$\log\left(\frac{\det \hat{\boldsymbol{A}}}{\det \boldsymbol{A}}\right) = \log(\kappa_{\mathrm{Kap}})$
Determinant, iterative solver	$\mathbb{E}\left\lvert \log\left(\frac{\mathrm{e}^{s_t} \det \hat{\boldsymbol{A}}}{\det \boldsymbol{A}}\right)\right\rvert^2 \leq \frac{4\log(\kappa_{\mathrm{Kap}})}{t}$

Table: Error bounds for direct and iterative solvers, assuming $\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^+) = \mathrm{rank}(\boldsymbol{A})$. Axelsson and Kaporin derived bound 2 for even t and we extended it. We derived bounds 1 and 4.

Conclusions and prospects

Structure of the Vecchia approximation

+ The Vecchia approximation is a superset of other approximations and has broad applicability.
+ We want to extend it to hierarchical matrices next. [Let me know if you're interested]

Optimality theory

+ Vecchia is optimal for any given sparsity pattern and permutation, but finding the best sparsity pattern and permutation is NP-hard.
+ Our paper surveys simple heuristics for the sparsity pattern and permutation.
+ There's a lot more work to build up sophisticated sparsity patterns and prove recovery guarantees.

Thank you for your attention! Does anyone have questions?