Tutorial: Foundations of Randomized Numerical Linear Algebra

Michał Dereziński

University of Michigan

Midwest RLA Workshop, May 11, 2026

Foundations of RandNLA

1. Fundamentals of sketching
   - The JL lemma
   - Subspace embeddings
   - Sketching least squares
   - Spectral approximation
2. Low-rank approximation
   - Randomized SVD
   - Regularized spectral approximation
   - Power method
3. Solving linear systems
   - Sketch-and-Precondition
   - Sketch-and-Project
   - Recursive preconditioning

NLA vs RandNLA

Task: Compute $\mathbf{x}^* = \mathrm{argmin}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ (least squares)

Numerical Linear Algebra (NLA):

Compute $\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b}$ *in* $O(nd^2)$ *time*

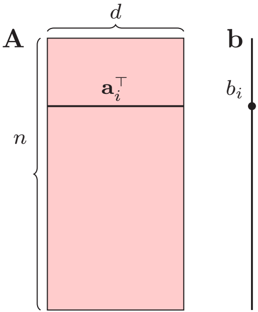
Randomized NLA (RandNLA):

Compute $\tilde{\mathbf{x}}$ such that

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq 1.01\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|$$

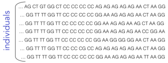
with prob. ≥ 0.99.

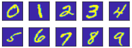
But it will be close to $O(nd)$ time!



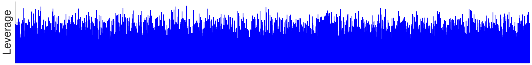
The RandNLA paradigm

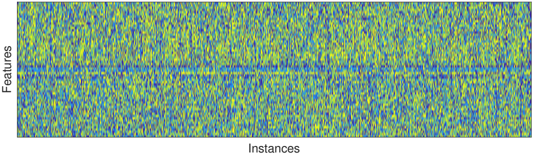
Genetics | Images | Location

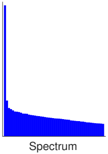




Domain:
Custom format







Raw data:
Matrices

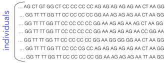
Algorithms:

Matrix sketch

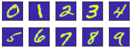
Approximation | Optimization | Visualization

The RandNLA paradigm

Genetics



Images



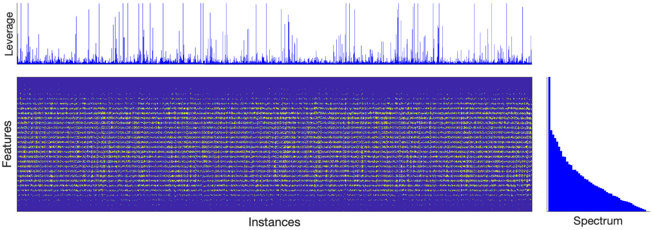
Location

Domain:

Custom format

Raw data:

Matrices



Algorithms:

Matrix sketch

Approximation

Optimization

Visualization

The RandNLA paradigm

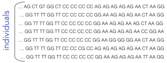
Genetics

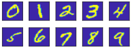
Images

Location

Domain:

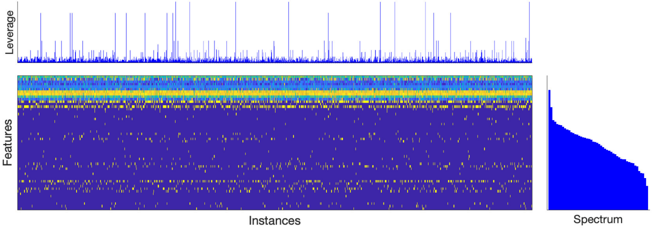
Custom format





Raw data:

Matrices



Algorithms:

Matrix sketch

Approximation

Optimization

Visualization

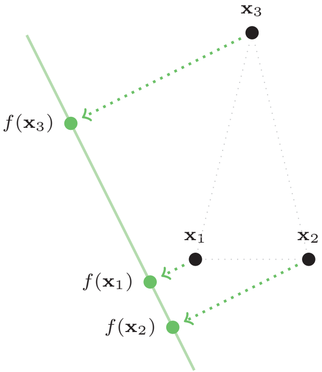
Outline

1. Fundamentals of sketching
   - The JL lemma
   - Subspace embeddings
   - Sketching least squares
   - Spectral approximation
2. Low-rank approximation
   - Randomized SVD
   - Regularized spectral approximation
   - Power method
3. Solving linear systems
   - Sketch-and-Precondition
   - Sketch-and-Project
   - Recursive preconditioning

Main goal: Dimensionality reduction

Given: Set of high-dimensional vectors $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subseteq \mathbb{R}^n$
Goal: Construct an embedding $f : \mathbb{R}^n \to \mathbb{R}^m$ with $m \ll n$ and

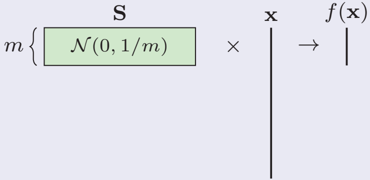
$$\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\| = (1 \pm \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|.$$



The JL lemma

Johnson & Lindenstrauss, Contemp. Math. 1984, [AV, FOCS99]

Consider a random linear embedding $f(\mathbf{x}) = \mathbf{S}\mathbf{x}$, where $\mathbf{S}$ is an $m \times n$ matrix with i.i.d. $\mathcal{N}(0, 1/m)$ Gaussian entries:

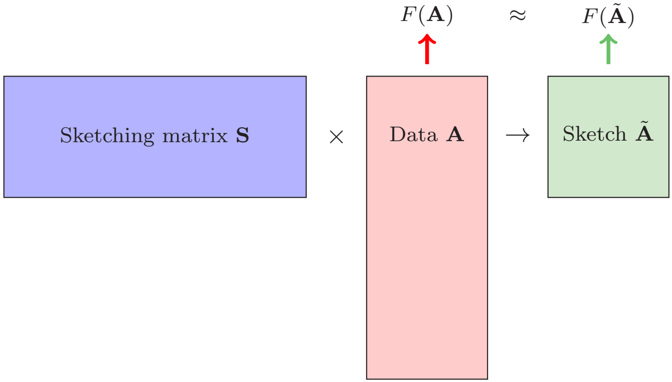


For $m = O(\log(N/\delta)/\epsilon^2)$ and all size N point sets $\mathcal{D} \subseteq \mathbb{R}^n$,

$$\Pr\Big(\|\mathbf{S}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{x}\| \quad \forall_{\mathbf{x}\in\mathcal{D}}\Big) \geq 1 - \delta. \qquad (\epsilon, \delta)\text{-embedding}$$

Matrix sketching

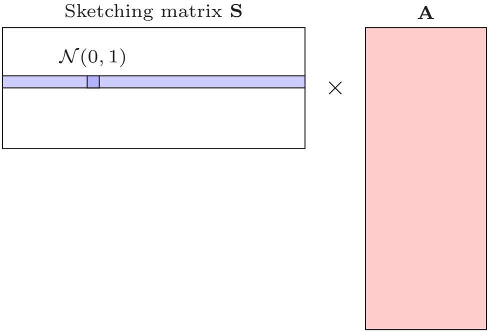
Sketching: Linear transformation (encoded as matrix $\mathbf{S}$) that reduces the dimensionality of a data matrix or a vector



Uniform sampling, Importance sampling, Fast Fourier transforms, Gaussian embeddings, Sparse embeddings, ...

Gaussian sketch

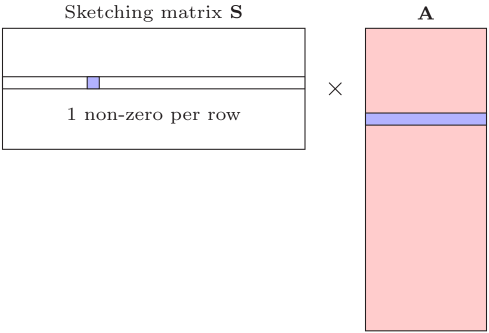
Sketching matrix $\mathbf{S}$ has i.i.d. Gaussian entries



Extension: Sparse matrices with random ± 1 entries

Subsampling sketch

Sketching matrix $\mathbf{S}$ randomly selects rows of $\mathbf{A}$



Extension: Data-aware importance sampling, e.g., using leverage scores

Subspace embeddings

Question: What if the dataset $\mathcal{D} \subseteq \mathbb{R}^n$ in the JL Lemma is infinite?

- Intuition: Possible if $\mathcal{D}$ has some low-dimensional structure
- Key example: $\mathcal{D}$ is a d-dimensional subspace of $\mathbb{R}^n$, $d \ll n$

Subspace embedding [Sar06]

Let $\mathcal{D}$ be a subspace of $\mathbb{R}^n$. $\mathbf{S}$ is an (ϵ, δ)-SE for $\mathcal{D}$ if

$$\|\mathbf{S}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{x}\| \quad \forall_{\mathbf{x}\in\mathcal{D}} \quad \text{with probability } 1-\delta.$$

Intuition: To obtain an effective "sketch" $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$,
we need an embedding for $\mathrm{range}(\mathbf{A}) = \{\mathbf{z} \mid \mathbf{z} = \mathbf{A}\mathbf{x}\}$

$$\|\mathbf{S}\mathbf{A}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\| \qquad \text{for all } \mathbf{x}$$

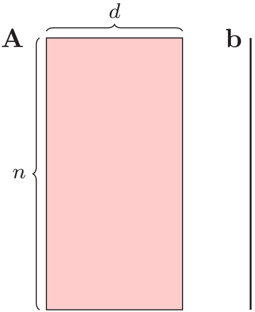
Sarlos. *"Improved approximation algorithms for large matrices via random projections."* FOCS 2006.

Example: Sketching least squares

Tall least squares problem:

$$\text{Compute} \quad \mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}} \underbrace{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2}_{L(\mathbf{x})}$$

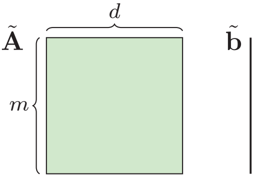
$$\text{for} \quad \mathbf{A} \in \mathbb{R}^{n \times d}, \ \mathbf{b} \in \mathbb{R}^n$$



Sketched least squares problem:

$$\text{Compute} \quad \tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \underbrace{\|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|^2}_{\tilde{L}(\mathbf{x})}$$

$$\text{for} \quad \underbrace{\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}}_{\text{sketch of } \mathbf{A}}, \ \underbrace{\tilde{\mathbf{b}} = \mathbf{S}\mathbf{b}}_{\text{sketch of } \mathbf{b}}$$



Example: Sketching least squares

Goal: Ensure that sketched objective $\tilde{L}(\mathbf{x})$ approximates $L(\mathbf{x})$?

Example: Sketching least squares

Goal: Ensure that sketched objective $\tilde{L}(\mathbf{x})$ approximates $L(\mathbf{x})$?

Idea: L is the norm of a vector in a $d+1$-dim. subspace:

$$L(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\| = \left\| \left[\mathbf{A} \,\middle|\, \mathbf{b} \right] \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|$$

Example: Sketching least squares

Goal: Ensure that sketched objective $\tilde{L}(\mathbf{x})$ approximates $L(\mathbf{x})$?

Idea: L is the norm of a vector in a $d+1$-dim. subspace:

$$L(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\| = \left\| \begin{bmatrix} \mathbf{A} & \big| & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|$$

Conclusion: If $\mathbf{S}$ is a ϵ-SE for $[\mathbf{A}|\mathbf{b}]$, then

$$\underbrace{\|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}\|}_{\tilde{L}(\mathbf{x})} = (1 \pm \epsilon)\underbrace{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|}_{L(\mathbf{x})} \qquad \text{for all } \mathbf{x} \in \mathbb{R}^d.$$

Sketched least squares guarantee

If $\mathbf{S}$ is a ϵ-SE for $[\mathbf{A}|\mathbf{b}]$, then $\tilde{\mathbf{x}} = \mathrm{argmin}_{\mathbf{x}} \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}\|$ satisfies

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \le \big(1 + O(\epsilon)\big)\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|.$$

Example: Sketching least squares

Gaussian subspace embedding

Let $\mathbf{S}$ be an $m \times n$ matrix with i.i.d. $\mathcal{N}(0, 1/m)$ entries. If $m = O(d/\epsilon^2)$, then $\mathbf{S}$ is an (ϵ, δ)-SE for any d-dimensional $\mathcal{D}$.

Conclusion: Sketched least squares error with Gaussian sketch

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|^2 \leq \Big(1 + O\big(\sqrt{d/m}\,\big)\Big)\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|^2.$$

Question: Is the subspace embedding analysis accurate?

Answer: Not really. In fact, for Gaussian sketch we can show:

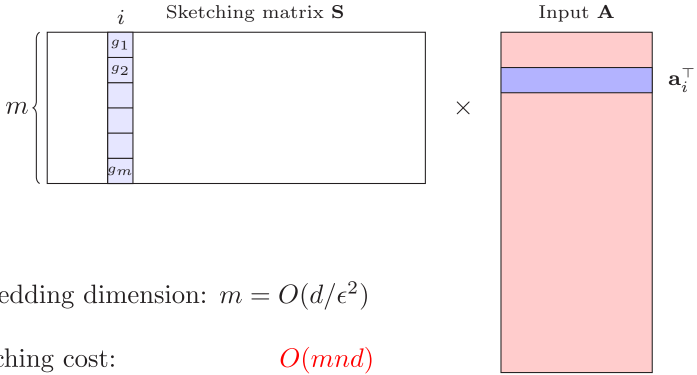
$$\mathbb{E}\,\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|^2 = \Big(1 + \frac{d}{m-d-1}\Big)\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|^2.$$

Extends to other sketches via Random Matrix Theory (RMT). [Der22]

[Der22] Dereziński. *"Algorithmic Gaussianization through Sketching: Converting Data into Sub-gaussian Random Designs"*. Conference on Learning Theory, 2023.

Making sketching fast

Gaussian sketch



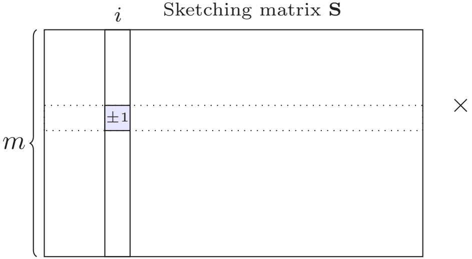
Embedding dimension: $m = O(d/\epsilon^2)$

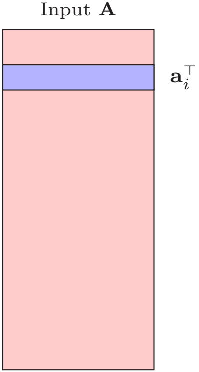
Sketching cost: $O(mnd)$

[CW17] Clarkson and Woodruff. *"Low-rank approximation and regression in input sparsity time."* JACM 2017.

[CDD26] Chenakkod, Dereziński, and Dong. *"Optimal subspace embeddings: Resolving nelson-nguyen conjecture up to sub-polylogarithmic factors."* SODA 2026.

CountSketch [CW17]





Embedding dimension: $m = O(d^2/\epsilon^2)$

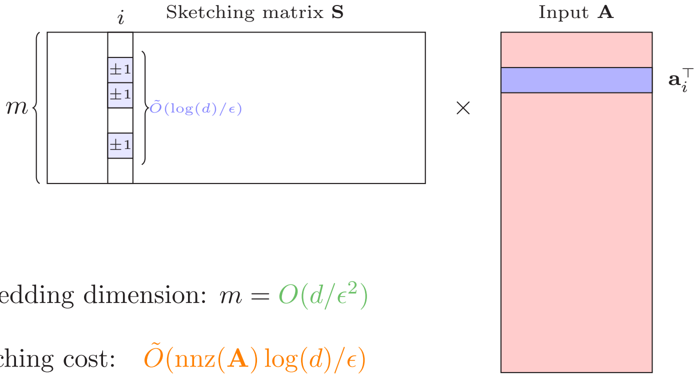
Sketching cost: $O(\mathrm{nnz}(\mathbf{A})) = O(nd)$

[CW17] Clarkson and Woodruff. *"Low-rank approximation and regression in input sparsity time."* JACM 2017.

[CDD26] Chenakkod, Dereziński, and Dong. *"Optimal subspace embeddings: Resolving nelson-nguyen conjecture up to sub-polylogarithmic factors."* SODA 2026.

Making sketching fast

<u>Sparse Embedding</u> (e.g., [NN13, Coh16, CDD26])



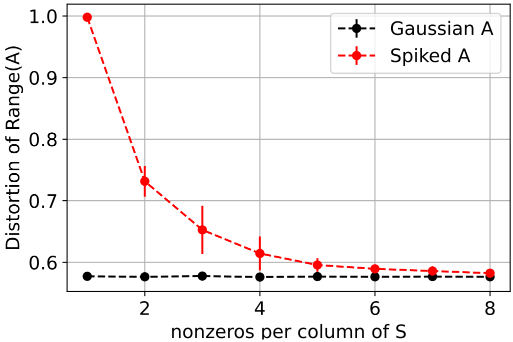
Embedding dimension: $m = O(d/\epsilon^2)$

Sketching cost: $\tilde{O}(\mathrm{nnz}(\mathbf{A}) \log(d)/\epsilon)$

[CW17] Clarkson and Woodruff. *"Low-rank approximation and regression in input sparsity time."* JACM 2017.

[CDD26] Chenakkod, Dereziński, and Dong. *"Optimal subspace embeddings: Resolving nelson-nguyen conjecture up to sub-polylogarithmic factors."* SODA 2026.

Effect of sparsity on embedding distortion



Embedding distortion ϵ of a sparse matrix $\mathbf{S}$ for range($\mathbf{A}$), where $\mathbf{A} \in \mathbb{R}^{10^5 \times 2 \cdot 10^3}$ and $m = 3d$. The "Spiked" $\mathbf{A}$ is formed by stacking the identity matrices and scaling random rows by 10^4.

[MDM+23] Murray et al. "Randomized Numerical Linear Algebra: A Perspective on the Field with an Eye to Software," *arXiv:2302.11474*, 2023.

Oblivious subspace embeddings

Oblivious subspace embeddings

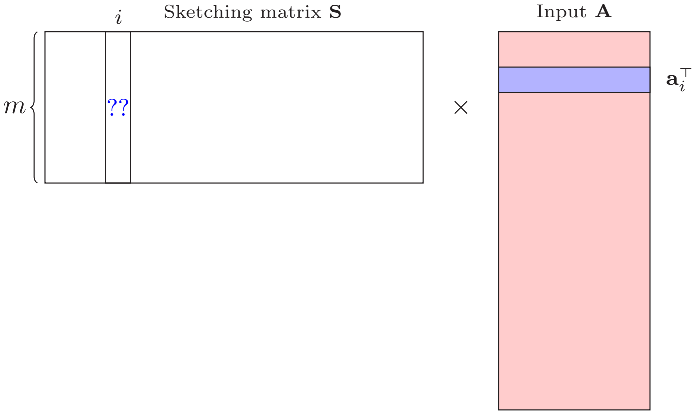
The following distributions over a sketching matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ are (ϵ,δ)-SE for *any* d-dimensional subspace $\mathcal{D}$ of $\mathbb{R}^n$:

1. $\mathbf{S}$ has independent sub-Gaussian entries scaled by $1/\sqrt{m}$, e.g., $\mathbf{S}_{ij} \sim \mathcal{N}(0, 1/r)$ or $\mathbf{S}_{ij} \sim \pm 1/\sqrt{r}$, and $m \geq O(d/\epsilon^2)$.
2. $\mathbf{S}$ is a sparse matrix with $s = \tilde{O}(\log(d)/\epsilon)$ random non-zero entries per column and $m \geq O(d/\epsilon^2)$.
3. $\mathbf{S} = \frac{1}{\sqrt{m}}\mathbf{I}_{S,:}\mathbf{H}\mathbf{D}$, where $\mathbf{D}$ is diagonal with random ± 1 entries, $\mathbf{H}$ is a Hadamard matrix, and S is a uniformly random subset of $\{1, ..., n\}$ with size $m \geq O(d\log(d)/\epsilon^2)$.

Data-aware subspace embeddings

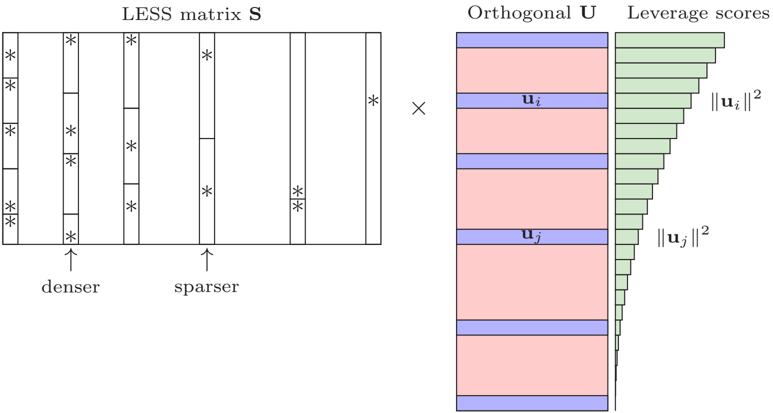
What if we want an embedding for a specific subspace $\mathcal{D}$?

Example: $\mathcal{D} = \mathrm{range}(\mathbf{A})$ for a given matrix $\mathbf{A}$



Can we further reduce the sparsity of $\mathbf{S}$ *by knowing* $\mathbf{A}$*?*

Leverage Score Sparsification (LESS)



Definition: Let $\mathbf{U}$ be the orthonormal basis for range($\mathbf{A}$).
The ith leverage score is the squared norm of the ith row of $\mathbf{U}$.

[DLDM21] Dereziński, Liao, Dobriban, and Mahoney. *"Sparse sketches with small inversion bias,"* COLT 2021.

Leverage score sparsified subspace embedding

Theorem ([CDD25])

Let $\mathbf{U}$ *be an* $n \times d$ *orthogonal matrix with rows* $\mathbf{u}_1, ..., \mathbf{u}_n$, *and let* $\mathbf{S}$ *be an* $m \times n$ *matrix with* nnz_i *non-zeros in the ith column. If*

$$(LESS) \qquad m \geq O(d/\epsilon^2) \quad \textit{and} \quad \mathrm{nnz}_i \geq \max\left\{ 1,\ \tilde{O}(\|\mathbf{u}_i\|^2/\epsilon) \right\}$$

then with high probability $\mathbf{S}$ *is an embedding for* $\mathrm{range}(\mathbf{U})$.

[CDD25] Chenakkod, Dereziński, and Dong. *"Optimal oblivious subspace embeddings with near-optimal sparsity,"* ICALP 2025.

Leverage score sparsified subspace embedding

Theorem ([CDD25])

Let $\mathbf{U}$ *be an* $n \times d$ *orthogonal matrix with rows* $\mathbf{u}_1, ..., \mathbf{u}_n$, *and let* $\mathbf{S}$ *be an* $m \times n$ *matrix with* nnz_i *non-zeros in the ith column. If*

$$(LESS) \qquad m \geq O(d/\epsilon^2) \quad \textit{and} \quad \mathrm{nnz}_i \geq \max\left\{1,\ \tilde{O}(\|\mathbf{u}_i\|^2/\epsilon)\right\}$$

then with high probability $\mathbf{S}$ *is an embedding for* $\mathrm{range}(\mathbf{U})$.

- Since $\sum_i \|\mathbf{u}_i\|^2 = d$, the total number of non-zeros in $\mathbf{S}$ is

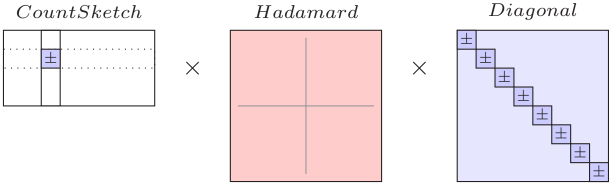
$$\underbrace{n + \tilde{O}(d/\epsilon)}_{\text{LESS}} \quad \ll \quad \underbrace{\tilde{O}\big(n\log(d)/\epsilon\big)}_{\text{data-oblivious}}$$

[CDD25] Chenakkod, Dereziński, and Dong. *"Optimal oblivious subspace embeddings with near-optimal sparsity,"* ICALP 2025.

Implication: CountSketch × RHT

Corollary (CountSketch × RHT)

For large enough n, "CountSketch × RHT" with $m = O(d/\epsilon^2)$ is an (ϵ, δ)-embedding for any d-dimensional subspace.



Proof. RHT flattens the leverage scores so that $\|\mathbf{u}_i\|^2 = O(d/n)$.

$$\text{nnz-per-col} \;=\; \max\left\{1,\ \tilde{O}(\|\mathbf{u}_i\|^2/\epsilon)\right\} \;=\; 1 \quad \text{for all } i.$$

[CDD25] Chenakkod, Dereziński, and Dong. *"Optimal oblivious subspace embeddings with near-optimal sparsity,"* ICALP 2025.

[CFS21] Cartis, Fiala, and Shao. *"Hashing embeddings of optimal dimension, with applications to linear least squares"*, arXiv:2105.11815, 2021.

Why is subspace embedding (too) powerful?

Example: Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a sketch of $\mathbf{A}$, and suppose that $\mathbf{S}$ is a subspace embedding for $\mathrm{range}(\mathbf{A}) = \{\mathbf{z} \mid \mathbf{z} = \mathbf{A}\mathbf{x}\}$

Why is subspace embedding (too) powerful?

Example: Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a sketch of $\mathbf{A}$, and suppose that $\mathbf{S}$ is a subspace embedding for $\mathrm{range}(\mathbf{A}) = \{\mathbf{z} \mid \mathbf{z} = \mathbf{A}\mathbf{x}\}$

$$\|\tilde{\mathbf{A}}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\| \qquad \text{for all } \mathbf{x}$$

Why is subspace embedding (too) powerful?

Example: Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a sketch of $\mathbf{A}$, and suppose that $\mathbf{S}$ is a subspace embedding for $\mathrm{range}(\mathbf{A}) = \{\mathbf{z} \mid \mathbf{z} = \mathbf{A}\mathbf{x}\}$

$$\|\tilde{\mathbf{A}}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\| \qquad \text{for all } \mathbf{x}$$

$$\Rightarrow \qquad \mathbf{x}^\top \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}\mathbf{x} = (1 \pm \epsilon)^2 \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} \qquad \text{for all } \mathbf{x}$$

Why is subspace embedding (too) powerful?

Example: Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a sketch of $\mathbf{A}$, and suppose that $\mathbf{S}$ is a subspace embedding for $\mathrm{range}(\mathbf{A}) = \{\mathbf{z} \mid \mathbf{z} = \mathbf{A}\mathbf{x}\}$

$$\|\tilde{\mathbf{A}}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\| \qquad \text{for all } \mathbf{x}$$

$$\Rightarrow \qquad \mathbf{x}^\top \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}\mathbf{x} = (1 \pm \epsilon)^2 \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} \qquad \text{for all } \mathbf{x}$$

$$\Rightarrow \qquad \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} = (1 \pm \epsilon)^2 \mathbf{A}^\top \mathbf{A} \qquad \text{(Löwner ordering)}$$

Why is subspace embedding (too) powerful?

Example: Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a sketch of $\mathbf{A}$, and suppose that $\mathbf{S}$ is a subspace embedding for $\mathrm{range}(\mathbf{A}) = \{\mathbf{z} \mid \mathbf{z} = \mathbf{A}\mathbf{x}\}$

$$\|\tilde{\mathbf{A}}\mathbf{x}\| = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\| \qquad \text{for all } \mathbf{x}$$

$$\Rightarrow \qquad \mathbf{x}^\top \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}\mathbf{x} = (1 \pm \epsilon)^2 \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} \qquad \text{for all } \mathbf{x}$$

$$\Rightarrow \qquad \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} = (1 \pm \epsilon)^2 \mathbf{A}^\top \mathbf{A} \qquad \text{(Löwner ordering)}$$

Spectral approximation

$\tilde{\mathbf{A}}$ is a ϵ-spectral approximation of $\mathbf{A}$ if $\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} = (1 \pm \epsilon)\mathbf{A}^\top \mathbf{A}$.

Tells us essentially everything about $\mathbf{A}^\top \mathbf{A}$ up to ϵ relative error, e.g.:

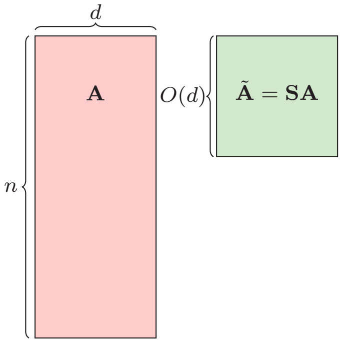
- $\sigma_i(\tilde{\mathbf{A}}) = (1 \pm \epsilon)\sigma_i(\mathbf{A})$ for every singular value.
- $\|\tilde{\mathbf{A}}\|_* = (1 \pm \epsilon)\|\mathbf{A}\|_*$ for every standard norm.

$\Rightarrow$ Probably overkill for most tasks (e.g., sketched least squares).

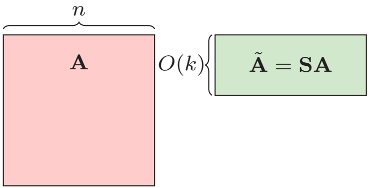
Outline

Two regimes of sketching

Rank-preserving sketch

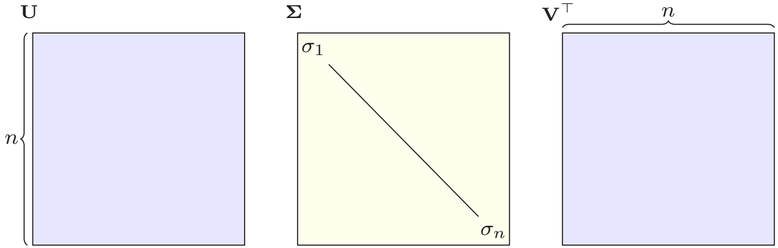


Low-rank sketch



Goal of low-rank approximation: Truncated SVD

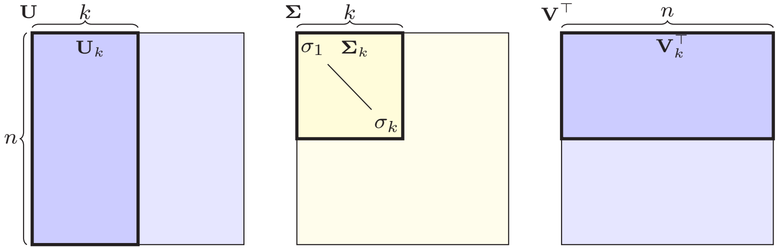
"Ideal" factorization of $\mathbf{A}$: Singular Value Decomposition



We order the singular values so that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$.

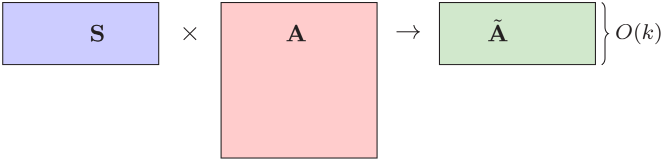
Goal of low-rank approximation: Truncated SVD

"Ideal" factorization of $\mathbf{A}$: Singular Value Decomposition



We order the singular values so that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$.

Randomized SVD



1. Form the right singular vector matrix $\mathbf{Q} \in \mathbb{R}^{n \times O(k)}$ of $\tilde{\mathbf{A}}$.
2. Compute SVD of the projected matrix $\mathbf{A}\mathbf{Q}\mathbf{Q}^\top = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^\top$.

Target: $\quad \|\mathbf{A} - \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^\top\| \approx \|\mathbf{A} - \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top\| = \sigma_{k+1}(\mathbf{A})$

[HMT11] Halko, Martinsson, and Tropp, "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, 53.2:217–288, 2011.

Failure of subspace embedding

Let $\mathbf{Q}$ be the right singular vectors of $\mathbf{SA}$.

Question: How to ensure a good low-rank approximation?

Failure of subspace embedding

Let $\mathbf{Q}$ be the right singular vectors of $\mathbf{SA}$.

Question: How to ensure a good low-rank approximation?

- What if sketch $\mathbf{S}$ is a subspace embedding for range($\mathbf{A}$)?

$$\Rightarrow \quad \mathbf{A}^\top\mathbf{S}^\top\mathbf{S}\mathbf{A} = (1 \pm \epsilon)\mathbf{A}^\top\mathbf{A} \quad \textit{(spectral approximation)}$$

Failure of subspace embedding

Let $\mathbf{Q}$ be the right singular vectors of $\mathbf{SA}$.

Question: How to ensure a good low-rank approximation?

- What if sketch $\mathbf{S}$ is a subspace embedding for range($\mathbf{A}$)?

$$\Rightarrow \quad \mathbf{A}^\top\mathbf{S}^\top\mathbf{S}\mathbf{A} = (1 \pm \epsilon)\mathbf{A}^\top\mathbf{A} \quad \textit{(spectral approximation)}$$

- This means that $\mathbf{A}\mathbf{Q}\mathbf{Q}^\top = \mathbf{A}$, regardless of ϵ!
- Only possible for sketch size $\geq$ rank($\mathbf{A}$) (way too big).

Failure of subspace embedding

Let $\mathbf{Q}$ be the right singular vectors of $\mathbf{SA}$.

Question: How to ensure a good low-rank approximation?

- What if sketch $\mathbf{S}$ is a subspace embedding for range($\mathbf{A}$)?

$$\Rightarrow \quad \mathbf{A}^\top\mathbf{S}^\top\mathbf{S}\mathbf{A} = (1 \pm \epsilon)\mathbf{A}^\top\mathbf{A} \quad \textit{(spectral approximation)}$$

- This means that $\mathbf{A}\mathbf{Q}\mathbf{Q}^\top = \mathbf{A}$, regardless of ϵ!
- Only possible for sketch size $\geq$ rank($\mathbf{A}$) (way too big).

Idea:

Spectral approximation that "ignores small singular values".

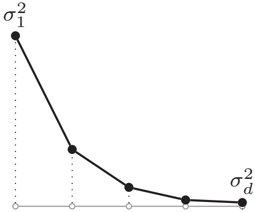
Regularized spectral approximation

Definition (e.g., [AM15, MM17])

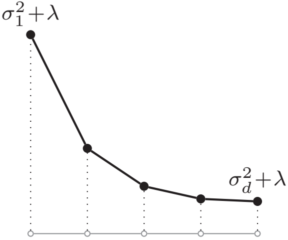
$\tilde{\mathbf{A}}$ is a λ-regularized ϵ-spectral approximation of $\mathbf{A}$ if:

$$\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} + \lambda \mathbf{I} = (1 \pm \epsilon)(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I}).$$

Spectrum of $\mathbf{A}^\top \mathbf{A}$



Spectrum of $\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I}$



A guarantee for Randomized SVD

Randomized SVD guarantee [CD26]

Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a λ-regularized ϵ-spectral approximation of $\mathbf{A}$. Let $\mathbf{Q}$ be the right singular vector matrix of $\tilde{\mathbf{A}}$. Then:

$$\|\mathbf{A} - \mathbf{A}\mathbf{Q}\mathbf{Q}^\top\|^2 \leq \frac{\lambda}{1-\epsilon}.$$

[CD26] Chenakkod and Dereziński. *"Accelerating power method with fast sketching for stronger low-rank approximation."* preprint forthcoming, 2026.

A guarantee for Randomized SVD

Randomized SVD guarantee [CD26]

Let $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ be a λ-regularized ϵ-spectral approximation of $\mathbf{A}$. Let $\mathbf{Q}$ be the right singular vector matrix of $\tilde{\mathbf{A}}$. Then:

$$\|\mathbf{A} - \mathbf{A}\mathbf{Q}\mathbf{Q}^\top\|^2 \leq \frac{\lambda}{1-\epsilon}.$$

Proof. Using the fact that $\mathbf{Q}\mathbf{Q}^\top = \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^\dagger$, we have

$$\begin{aligned}
\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top &= (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \lambda\mathbf{I})(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \lambda\mathbf{I})^{-1} - \mathbf{Q}\mathbf{Q}^\top \\
&= \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \lambda\mathbf{I})^{-1} - \mathbf{Q}\mathbf{Q}^\top + \lambda(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \lambda\mathbf{I})^{-1} \\
&\preceq \frac{\lambda}{1-\epsilon}(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})^{-1} \\
&\Downarrow
\end{aligned}$$

$$\|\mathbf{A}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top)\|^2 \leq \frac{\lambda}{1-\epsilon}\|\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^\top\| \leq \frac{\lambda}{1-\epsilon}.$$

[CD26] Chenakkod and Dereziński. *"Accelerating power method with fast sketching for stronger low-rank approximation."* preprint forthcoming, 2026.

A guarantee for Randomized SVD

Sketches with regularized spectral approximation

For all of the earlier "d-dimensional subspace embeddings" $\mathbf{S}$, w.h.p. $\mathbf{SA}$ is a λ-regularized ϵ-spectral approximation of $\mathbf{A}$ for:

$$\lambda = \frac{1}{d}\sum_{i>d}\sigma_i^2(\mathbf{A}).$$

Matches the standard Gaussian analysis [HMT11] up to constants.

A guarantee for Randomized SVD

Sketches with regularized spectral approximation

For all of the earlier "d-dimensional subspace embeddings" $\mathbf{S}$, w.h.p. $\mathbf{SA}$ is a λ-regularized ϵ-spectral approximation of $\mathbf{A}$ for:

$$\lambda = \frac{1}{d}\sum_{i>d}\sigma_i^2(\mathbf{A}).$$

$\Rightarrow$ Randomized SVD with, e.g., Gaussian $\mathbf{S}$ of size $O(k)$, satisfies:

$$\left\|\mathbf{A} - \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^\top\right\| \leq \sqrt{\frac{1}{k}\sum_{i>k}\sigma_i^2(\mathbf{A})} \leq \sqrt{n/k}\cdot\sigma_{k+1}(\mathbf{A}).$$

Question: What if we need a better approximation?

Matches the standard Gaussian analysis [HMT11] up to constants.

Refining Randomized SVD with Power Method

- Initialize $\mathbf{V}_0 \in \mathbb{R}^{n \times O(k)}$ (e.g., Gaussian matrix)
- Repeat $\mathbf{V}_{i+1} = \mathit{orthonormalize}(\mathbf{A}^\top \mathbf{A} \mathbf{V}_i)$
- Return $\mathbf{Q} = \mathbf{V}_q$

Lemma ([Woo14])

For any orthonormal $\mathbf{Q}$,

$$\|\mathbf{A} - \mathbf{A}\mathbf{Q}\mathbf{Q}^\top\| \leq \|(\mathbf{A}^\top\mathbf{A})^q - (\mathbf{A}^T\mathbf{A})^q\mathbf{Q}\mathbf{Q}^\top\|^{\frac{1}{2q}}$$

With $q = \mathcal{O}(\log(n)/\epsilon)$, *power method obtains* $\mathbf{Q}$ *such that w.h.p.*

$$\|\mathbf{A} - \mathbf{A}\mathbf{Q}\mathbf{Q}^\top\| \leq (1+\epsilon) \cdot \sigma_{k+1}(\mathbf{A}).$$

[HMT11, Woo14, MM15]

Outline

Solving linear systems with RandNLA

Task: Given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, solve $\mathbf{Ax} = \mathbf{b}$.

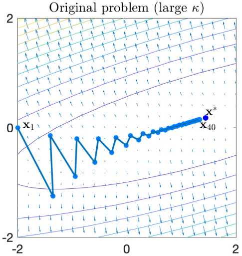
Sketch-and-Precondition: Construct a preconditioner by sketching matrix $\mathbf{A}$, and then run your favorite iterative solver.

- *"Least squares" approach:* Use a subspace embedding of $\mathbf{A}$
  Applicable only when the system is very "tall" (over-determined) or very "wide" (under-determined). [AMT10, MSM14]
- *"Low-rank" approach:* Use a low-rank approximation of $\mathbf{A}$
  Applicable for "low-rank+noise" matrices $\mathbf{A}$ or with regularized linear systems, i.e., where $\mathbf{A} = \mathbf{K} + \lambda\mathbf{I}$. [ACW17, FTU21]

Sketch-and-Project: Iteratively solve sketched systems. [GR15, DY24]

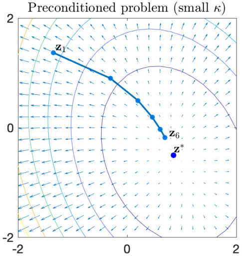
Sketch-and-Precondition for tall least squares

1. Rewrite least squares via normal equations, $\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b}$
2. Precondition an iterative method with $\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} = (1 \pm \frac{1}{2})\mathbf{A}^\top\mathbf{A}$



$$\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b}$$

$\kappa(\mathbf{A}^\top\mathbf{A})$ is large



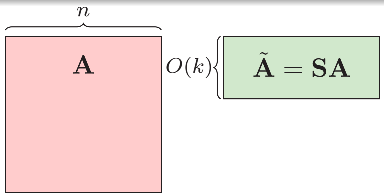
$$\mathbf{A}^\top\mathbf{A}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{z} = \mathbf{b}$$

$\kappa(\mathbf{A}^\top\mathbf{A}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1})$ is small

---

Rokhlin and Tygert, “A fast randomized algorithm for overdetermined linear least-squares regression”, *Proceedings of the National Academy of Sciences*, 2008.

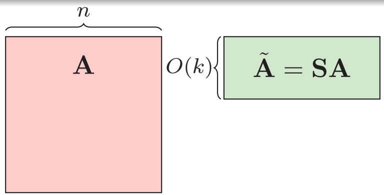
Does this work for general linear systems?



Does this work for general linear systems?

"Low-rank" approach. Precondition using $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{O(k)\times n}$. But then:

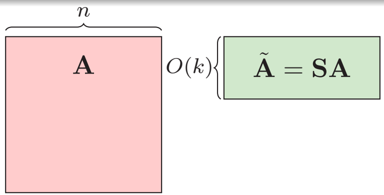
$$\sigma_i^2(\tilde{\mathbf{A}}) = (1 \pm \tfrac{1}{2})\sigma_i^2(\mathbf{A}) \pm \underbrace{\frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A})}_{\text{tail noise}}$$



Does this work for general linear systems?

"Low-rank" approach. Precondition using $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{O(k)\times n}$. But then:

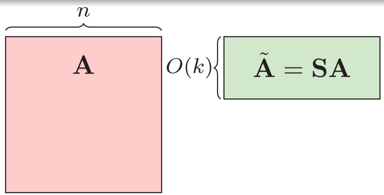
$$\sigma_i^2(\tilde{\mathbf{A}}) = (1 \pm \tfrac{1}{2})\sigma_i^2(\mathbf{A}) \pm \underbrace{\frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A})}_{\text{tail noise}}$$



Bottom line: The sketch picks up **noise** from the **spectral tail**
May be good enough (e.g., for some psd systems), [ACW17, FTU21]
but not very effective in general.

Does this work for general linear systems?

"Low-rank" approach. Precondition using $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{O(k)\times n}$. But then:



$$\sigma_i^2(\tilde{\mathbf{A}}) = (1 \pm \tfrac{1}{2})\sigma_i^2(\mathbf{A}) \pm \underbrace{\frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A})}_{\text{tail noise}}$$

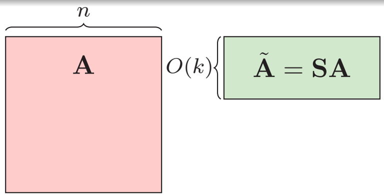
Bottom line: The sketch picks up **noise** from the **spectral tail**
May be good enough (e.g., for some psd systems), [ACW17, FTU21]
but not very effective in general.

Can we improve on this basic strategy?

Does this work for general linear systems?

"Low-rank" approach. Precondition using $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{O(k)\times n}$. But then:

$$\sigma_i^2(\tilde{\mathbf{A}}) = (1 \pm \tfrac{1}{2})\sigma_i^2(\mathbf{A}) \pm \underbrace{\frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A})}_{\text{tail noise}}$$



Bottom line: The sketch picks up **noise** from the **spectral tail**
May be good enough (e.g., for some psd systems), [ACW17, FTU21]
but not very effective in general.

Can we improve on this basic strategy?

Yes! By using multiple sketches:

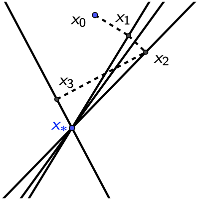
1. Sketch-and-Project
2. Recursive preconditioning

Background: The Kaczmarz algorithm

Idea: Iteratively project onto the solutions of individual equations.

Starting at $\mathbf{x}_0$, for $t = 0, 1, 2, ...$

1. Select index I_t
2. Project current iterate $\mathbf{x}_t$ onto the solutions of I_t-th equation



Randomized Kaczmarz: Select indices via weighted sampling [SV09]

The first Kaczmarz algorithm with provable convergence rate.

[Kac37] Stefan Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen", *Bulletin International de l'Académie Polonaise des Sciences et des Lettres* 35:355–357, 1937.

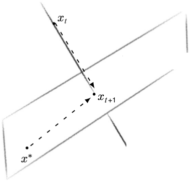
[SV09] Strohmer and Vershynin, "A randomized Kaczmarz algorithm with exponential convergence", *Journal of Fourier Analysis and Applications*, 14.2:262-278, 2009.

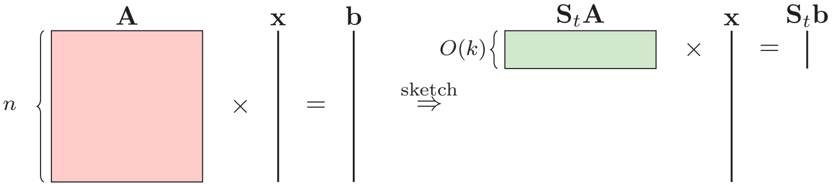
Powerful extension: Sketch-and-Project

Starting at $\mathbf{x}_0 \in \mathbb{R}^n$, for $t = 0, 1, 2, \ldots$

1. Sample random $O(k) \times n$ matrix $\mathbf{S}_t$.
2. Project $\mathbf{x}_t$ onto the solutions of $\mathbf{S}_t\mathbf{A}\mathbf{x} = \mathbf{S}_t\mathbf{b}$:

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}_t - \mathbf{x}\|^2 \quad \text{subject to} \quad \mathbf{S}_t\mathbf{A}\mathbf{x} = \mathbf{S}_t\mathbf{b}.$$





[GR15] Gower and Richtárik, "Randomized iterative methods for linear systems", *SIAM Journal on Matrix Analysis and Applications*, 36.4:1660-1690, 2015.

Key insight: RMT analysis of Sketch-and-Project

Theorem ([DR22])

If $\mathbf{S}_t \in \mathbb{R}^{O(k)\times n}$ *is Gaussian, then Sketch-and-Project satisfies:*

$$\mathbb{E}\,\|\mathbf{x}_t - \mathbf{x}^*\|^2 \;\leq\; \left(1 - \frac{\sigma_{\min}^2(\mathbf{A})}{\frac{1}{k}\sum_{i>k}\sigma_i^2(\mathbf{A})}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

[DR22] Dereziński and Rebrova, "Sharp Analysis of Sketch-and-Project Methods via a Connection to Randomized Singular Value Decomposition", *SIAM Journal on Mathematics of Data Science*, 6.1:127-153, 2024.

Key insight: RMT analysis of Sketch-and-Project

Theorem ([DR22])

If $\mathbf{S}_t \in \mathbb{R}^{O(k)\times n}$ *is Gaussian, then Sketch-and-Project satisfies:*

$$\mathbb{E}\,\|\mathbf{x}_t - \mathbf{x}^*\|^2 \;\leq\; \left(1 - \frac{\sigma_{\min}^2(\mathbf{A})}{\frac{1}{k}\sum_{i>k}\sigma_i^2(\mathbf{A})}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

We again observe the ***tail noise*** *from sketching.*

[DR22] Dereziński and Rebrova, "Sharp Analysis of Sketch-and-Project Methods via a Connection to Randomized Singular Value Decomposition", *SIAM Journal on Mathematics of Data Science*, 6.1:127-153, 2024.

Key insight: RMT analysis of Sketch-and-Project

Theorem ([DR22])

If $\mathbf{S}_t \in \mathbb{R}^{O(k)\times n}$ is Gaussian, then Sketch-and-Project satisfies:

$$\mathbb{E}\,\|\mathbf{x}_t - \mathbf{x}^*\|^2 \;\leq\; \left(1 - \frac{\sigma^2_{\min}(\mathbf{A})}{\frac{1}{k}\sum_{i>k}\sigma_i^2(\mathbf{A})}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

We again observe the ***tail noise*** *from sketching.*

Key insight:

- Each sketch-and-project step runs on a $\frac{k}{n}$**-fraction** of the data
- This runtime gain should cancel out the **tail noise** $\leq \frac{n}{k}\sigma^2_{k+1}$

[DR22] Dereziński and Rebrova, "Sharp Analysis of Sketch-and-Project Methods via a Connection to Randomized Singular Value Decomposition", *SIAM Journal on Mathematics of Data Science*, 6.1:127-153, 2024.

Key insight: RMT analysis of Sketch-and-Project

Theorem ([DR22])

If $\mathbf{S}_t \in \mathbb{R}^{O(k)\times n}$ is Gaussian, then Sketch-and-Project satisfies:

$$\mathbb{E}\,\|\mathbf{x}_t - \mathbf{x}^*\|^2 \;\le\; \left(1 - \frac{\sigma_{\min}^2(\mathbf{A})}{\frac{1}{k}\sum_{i>k}\sigma_i^2(\mathbf{A})}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

We again observe the ***tail noise*** *from sketching.*

Key insight:

- Each sketch-and-project step runs on a $\frac{\cancel{k}}{\cancel{n}}$-**fraction** of the data
- This runtime gain should cancel out the **tail noise** $\le \frac{\cancel{n}}{\cancel{k}}\sigma_{k+1}^2$

[DR22] Dereziński and Rebrova, "Sharp Analysis of Sketch-and-Project Methods via a Connection to Randomized Singular Value Decomposition", *SIAM Journal on Mathematics of Data Science*, 6.1:127-153, 2024.

Key insight: RMT analysis of Sketch-and-Project

Theorem ([DR22])

If $\mathbf{S}_t \in \mathbb{R}^{O(k)\times n}$ is Gaussian, then Sketch-and-Project satisfies:

$$\mathbb{E}\,\|\mathbf{x}_t - \mathbf{x}^*\|^2 \;\leq\; \left(1 - \frac{\sigma^2_{\min}(\mathbf{A})}{\frac{1}{k}\sum_{i>k}\sigma_i^2(\mathbf{A})}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

*We again observe the **tail noise** from sketching.*

Key insight:

- Each sketch-and-project step runs on a $\cancel{k}/\cancel{n}$-**fraction** of the data
- This runtime gain should cancel out the **tail noise** $\leq \frac{\cancel{n}}{\cancel{k}}\sigma^2_{k+1}$

Hang on, Gaussian sketching is still too expensive!

[DR22] Dereziński and Rebrova, "Sharp Analysis of Sketch-and-Project Methods via a Connection to Randomized Singular Value Decomposition", *SIAM Journal on Mathematics of Data Science*, 6.1:127-153, 2024.

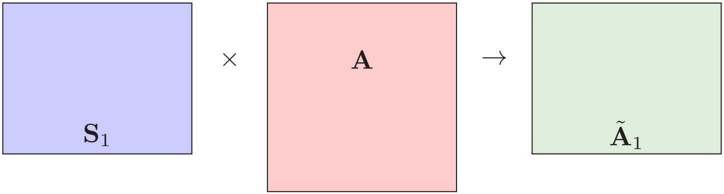
Making this efficient

Advances in sketch-and-project for systems with low-rank structure:

1. Sketching: *Gaussian guarantees for fast sketches*
   - Subsampled randomized Hadamard transform [DY24]
   - Leverage score sampling [RFY$^+$25]
2. Projecting: *Fast computation of the projection step*
   - Fast inner solver using PCG [DY24]
   - Amortizing projection cost across iterations [DNRY25]
3. Acceleration: *Improved convergence using momentum*
   - Convergence analysis with Nesterov's momentum [DLNR24]
   - Adaptive tuning of momentum parameters [DNRY25]

Dereziński and Yang, "Solving dense linear systems faster than via preconditioning", *56th Annual ACM Symposium on Theory of Computing*, 2024.

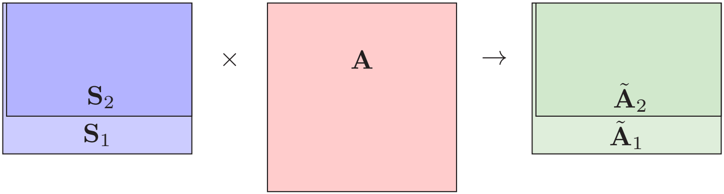
Different strategy: Recursive Preconditioning



1. Precondition $\mathbf{A}$ using large sketch $\tilde{\mathbf{A}}_1 = \mathbf{S}_1\mathbf{A}$,

Spielman and Teng, "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems", *SIAM Journal on Matrix Analysis and Applications*, 2014.

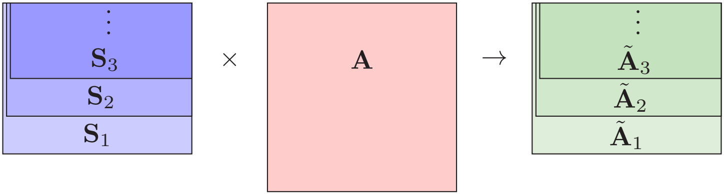
Different strategy: Recursive Preconditioning



1. Precondition $\mathbf{A}$ using large sketch $\tilde{\mathbf{A}}_1 = \mathbf{S}_1\mathbf{A}$,
2. Precondition $\tilde{\mathbf{A}}_1$ using smaller sketch $\tilde{\mathbf{A}}_2 = \mathbf{S}_2\mathbf{A}$,

Spielman and Teng, "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems", *SIAM Journal on Matrix Analysis and Applications*, 2014.

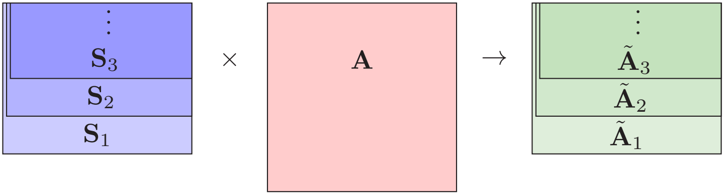
Different strategy: Recursive Preconditioning



1. Precondition $\mathbf{A}$ using large sketch $\tilde{\mathbf{A}}_1 = \mathbf{S}_1\mathbf{A}$,
2. Precondition $\tilde{\mathbf{A}}_1$ using smaller sketch $\tilde{\mathbf{A}}_2 = \mathbf{S}_2\mathbf{A}$,
3. Precondition $\tilde{\mathbf{A}}_2$ using even smaller sketch $\tilde{\mathbf{A}}_3 = \mathbf{S}_3\mathbf{A}$, etc.

Spielman and Teng, "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems", *SIAM Journal on Matrix Analysis and Applications*, 2014.

Different strategy: Recursive Preconditioning



1. Precondition $\mathbf{A}$ using large sketch $\tilde{\mathbf{A}}_1 = \mathbf{S}_1\mathbf{A}$,
2. Precondition $\tilde{\mathbf{A}}_1$ using smaller sketch $\tilde{\mathbf{A}}_2 = \mathbf{S}_2\mathbf{A}$,
3. Precondition $\tilde{\mathbf{A}}_2$ using even smaller sketch $\tilde{\mathbf{A}}_3 = \mathbf{S}_3\mathbf{A}$, etc.

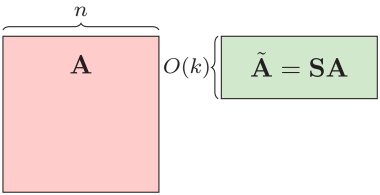
Domain-specific examples of recursive preconditioning techniques:

- Recursive solvers for graph Laplacians in theoretical computer science
- Multigrid solvers for differential equations in scientific computing

Spielman and Teng, "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems", *SIAM Journal on Matrix Analysis and Applications*, 2014.

Reminder: Single-sketch preconditioner fails

"Low-rank" approach. Precondition using $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{O(k)\times n}$. But then:



$$\sigma_i^2(\tilde{\mathbf{A}}) = (1 \pm \tfrac{1}{2})\sigma_i^2(\mathbf{A}) \pm \underbrace{\frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A})}_{\text{tail noise}}$$

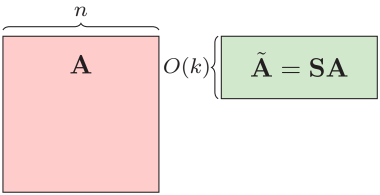
Bottom line: The sketch picks up **noise** from the **spectral tail**

$$\Rightarrow \qquad \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \not\approx \mathbf{A}^\top\mathbf{A}$$

Reminder: Single-sketch preconditioner fails

"Low-rank" approach. Precondition using $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{O(k)\times n}$. But then:

$$\sigma_i^2(\tilde{\mathbf{A}}) = (1 \pm \tfrac{1}{2})\sigma_i^2(\mathbf{A}) \pm \underbrace{\frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A})}_{\text{tail noise}}$$



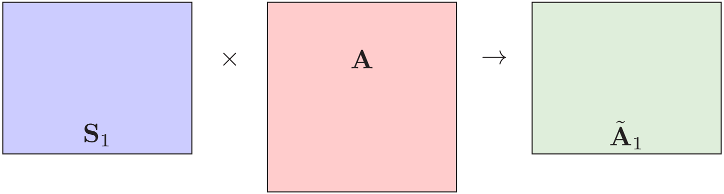
Bottom line: The sketch picks up **noise** from the **spectral tail**

$$\Rightarrow \qquad \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \not\approx \mathbf{A}^\top\mathbf{A}$$

$$\text{but} \qquad \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \lambda\mathbf{I} \approx \underbrace{\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I}}_{\text{more low-rank}}, \qquad \text{for} \quad \lambda = \frac{1}{k}\sum_{j>k}\sigma_j^2(\mathbf{A}).$$

Idea: Precondition with a regularized spectral approximation

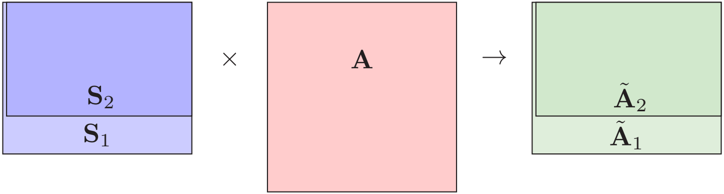
Strategy: Recursively impose more low-rank structure



$$
\begin{array}{ccc}
\mathbf{A}^\top \mathbf{A} \;\approx\; \mathbf{A}^\top \mathbf{A} + \sigma_{\min}^2 \mathbf{I} & \overbrace{O(\log(1/\epsilon))}^{\text{iters}} & \overbrace{\text{matvec}(\mathbf{A})}^{\text{cost per iter}} \\
\wr & \times & \\
\tilde{\mathbf{A}}_1^\top \tilde{\mathbf{A}}_1 + 2\sigma_{\min}^2 \mathbf{I} & \tilde{O}(1) & \text{matvec}(\tilde{\mathbf{A}}_1)
\end{array}
$$

[DS25] Dereziński and Sidford. *"Approaching Optimality for Solving Dense Linear Systems with Low-Rank Structure"*, SODA 2026.

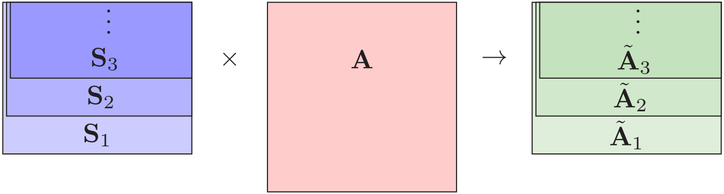
Strategy: Recursively impose more low-rank structure



$$
\begin{array}{cccc}
\mathbf{A}^\top\mathbf{A} \;\approx\; \mathbf{A}^\top\mathbf{A} + \sigma_{\min}^2\mathbf{I} & \overbrace{O(\log(1/\epsilon))}^{\text{iters}} & \overbrace{\mathrm{matvec}(\mathbf{A})}^{\text{cost per iter}} \\
\wr & \times & \\
\tilde{\mathbf{A}}_1^\top\tilde{\mathbf{A}}_1 + 2\sigma_{\min}^2\mathbf{I} & \tilde{O}(1) & \mathrm{matvec}(\tilde{\mathbf{A}}_1) \\
\wr & \times & \\
\tilde{\mathbf{A}}_2^\top\tilde{\mathbf{A}}_2 + 4\sigma_{\min}^2\mathbf{I} & \tilde{O}(1) & \mathrm{matvec}(\tilde{\mathbf{A}}_2)
\end{array}
$$

[DS25] Dereziński and Sidford. *"Approaching Optimality for Solving Dense Linear Systems with Low-Rank Structure"*, SODA 2026.

Strategy: Recursively impose more low-rank structure



$$\begin{array}{rcccc}
\mathbf{A}^\top\mathbf{A} & \approx & \mathbf{A}^\top\mathbf{A} + \sigma_{\min}^2\mathbf{I} & \overbrace{O(\log(1/\epsilon))}^{\text{iters}} & \overbrace{\mathrm{matvec}(\mathbf{A})}^{\text{cost per iter}} \\
 & & \wr\wr & \times & \\
 & & \tilde{\mathbf{A}}_1^\top\tilde{\mathbf{A}}_1 + 2\sigma_{\min}^2\mathbf{I} & \tilde{O}(1) & \mathrm{matvec}(\tilde{\mathbf{A}}_1) \\
 & & \wr\wr & \times & \\
 & & \tilde{\mathbf{A}}_2^\top\tilde{\mathbf{A}}_2 + 4\sigma_{\min}^2\mathbf{I} & \tilde{O}(1) & \mathrm{matvec}(\tilde{\mathbf{A}}_2) \\
 & & \wr\wr & \times & \\
 & & \tilde{\mathbf{A}}_3^\top\tilde{\mathbf{A}}_3 + 8\sigma_{\min}^2\mathbf{I} & \tilde{O}(1) & \mathrm{matvec}(\tilde{\mathbf{A}}_3)
\end{array}$$

[DS25] Dereziński and Sidford. *"Approaching Optimality for Solving Dense Linear Systems with Low-Rank Structure"*, SODA 2026.

Summary and conclusions

1. Fundamentals of sketching
   - The JL lemma
   - Subspace embeddings
   - Sketching least squares
   - Spectral approximation
2. Low-rank approximation
   - Randomized SVD
   - Regularized spectral approximation
   - Power method
3. Solving linear systems
   - Sketch-and-Precondition
   - Sketch-and-Project
   - Recursive preconditioning

References I

Haim Avron, Kenneth L Clarkson, and David P Woodruff.
Faster kernel ridge regression using sketching and preconditioning.
SIAM Journal on Matrix Analysis and Applications, 38(4):1116–1138, 2017.

Ahmed El Alaoui and Michael W. Mahoney.
Fast randomized kernel ridge regression with statistical guarantees.
In Proceedings of the 28th International Conference on Neural Information Processing Systems, pages 775–783, 2015.

Haim Avron, Petar Maymounkov, and Sivan Toledo.
Blendenpik: Supercharging lapack's least-squares solver.
SIAM Journal on Scientific Computing, 32(3):1217–1236, 2010.

Shabarish Chenakkod and Michał Dereziński.
Accelerating power method with fast sketching for stronger low-rank approximation.
arXiv preprint forthcoming, 2026.

Shabarish Chenakkod, Michał Dereziński, and Xiaoyu Dong.
Optimal oblivious subspace embeddings with near-optimal sparsity.
In Proceedings of the 52nd EATCS International Colloquium on Automata, Languages, and Programming, 2025.

Shabarish Chenakkod, Michał Dereziński, and Xiaoyu Dong.
Optimal subspace embeddings: Resolving nelson-nguyen conjecture up to sub-polylogarithmic factors.
ACM-SIAM Symposium on Discrete Algorithms, 2026.

Coralia Cartis, Jan Fiala, and Zhen Shao.
Hashing embeddings of optimal dimension, with applications to linear least squares.
arXiv preprint arXiv:2105.11815, 2021.

References II

Michael B Cohen.
Nearly tight oblivious subspace embeddings by trace inequalities.
In Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms, pages 278–287. SIAM, 2016.

Kenneth L. Clarkson and David P. Woodruff.
Low-rank approximation and regression in input sparsity time.
J. ACM, 63(6):54:1–54:45, January 2017.

Michał Dereziński.
Algorithmic gaussianization through sketching: Converting data into sub-gaussian random designs.
arXiv preprint arXiv:2206.10291, 2022.

Michał Dereziński, Zhenyu Liao, Edgar Dobriban, and Michael Mahoney.
Sparse sketches with small inversion bias.
In Conference on Learning Theory, pages 1467–1510. PMLR, 2021.

Michał Dereziński, Daniel LeJeune, Deanna Needell, and Elizaveta Rebrova.
Fine-grained analysis and faster algorithms for iteratively solving linear systems.
arXiv preprint arXiv:2405.05818, 2024.

Michał Dereziński, Deanna Needell, Elizaveta Rebrova, and Jiaming Yang.
Randomized kaczmarz methods with beyond-krylov convergence.
arXiv preprint arXiv:2501.11673, 2025.

Michał Dereziński and Elizaveta Rebrova.
Sharp analysis of sketch-and-project methods via a connection to randomized singular value decomposition.
arXiv preprint arXiv:2208.09585, 2022.

References III

Michał Dereziński and Aaron Sidford.
Approaching optimality for solving dense linear systems with low-rank structure.
arXiv preprint arXiv:2507.11724, 2025.

Michał Dereziński and Jiaming Yang.
Solving dense linear systems faster than via preconditioning.
In 56th Annual ACM Symposium on Theory of Computing, 2024.

Zachary Frangella, Joel A Tropp, and Madeleine Udell.
Randomized nystrom preconditioning.
arXiv preprint arXiv:2110.02820, 2021.

Robert M Gower and Peter Richtárik.
Randomized iterative methods for linear systems.
SIAM Journal on Matrix Analysis and Applications, 36(4):1660–1690, 2015.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp.
Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions.
SIAM review, 53(2):217–288, 2011.

M. S. Kaczmarz.
Angenaherte auflosung von systemen linearer gleichungen.
Bulletin International de l'Academie Polonaise des Sciences et des Lettres, 35:355–357, 1937.

Riley Murray, James Demmel, Michael W Mahoney, N Benjamin Erichson, Maksim Melnichenko, Osman Asif Malik, Laura Grigori, Piotr Luszczek, Michał Dereziński, Miles E Lopes, et al.
Randomized numerical linear algebra: A perspective on the field with an eye to software.
arXiv preprint arXiv:2302.11474, 2023.

References IV

Cameron Musco and Christopher Musco.
Randomized block krylov methods for stronger and faster approximate singular value decomposition.
Advances in neural information processing systems, 28, 2015.

Cameron Musco and Christopher Musco.
Recursive sampling for the nystrom method.
Advances in neural information processing systems, 30, 2017.

X. Meng, M. A. Saunders, and M. W. Mahoney.
LSRN: A parallel iterative solver for strongly over- or under-determined systems.
SIAM Journal on Scientific Computing, 36(2):C95–C118, 2014.

Jelani Nelson and Huy L. Nguyên.
Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings.
In Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS '13, pages 117–126, Washington, DC, USA, 2013. IEEE Computer Society.

Pratik Rathore, Zachary Frangella, Jiaming Yang, Michał Dereziński, and Madeleine Udell.
Have askotch: A neat solution for large-scale kernel ridge regression.
arXiv preprint arXiv:2407.10070, 2025.

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

Thomas Strohmer and Roman Vershynin.
A randomized kaczmarz algorithm with exponential convergence.
Journal of Fourier Analysis and Applications, 15(2):262–278, 2009.

References V

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