M\crNA
Manchester Numerical Analysis

Stefan Güttel

Randomized sketching for Krylov methods

MANCHESTER 1824

Remembering Olga & Jack

Olga Taussky-Todd (1906–1995)

John (Jack) Todd (1911–2007)

Photo in their Caltech office (1970s)

Randomized Numerical Linear Algebra

Randomized numerical linear algebra concerns the use of randomization as a resource to develop improved algorithms for large-scale linear algebra computations.

Murray et al.[1]

Randomized methods can handle certain NLA problems faster than any classical algorithm. Randomization allows us to tackle problems that otherwise seem impossible.

Martinsson-Tropp[2]

Our focus: Randomization + Krylov methods

[1] *Randomized Numerical Linear Algebra: A Perspective on the Field With an Eye to Software,* arXiv:2302.11474, 2023.

[2] *Randomized Numerical Linear Algebra: Foundations and Algorithms,* Acta Numerica, 2020.

Krylov methods are central to numerical linear algebra

Widely used for computations with large sparse or structured matrices, including:

- eigenvalue problems
- linear systems of equations
- least squares problems
- matrix functions
- matrix equations
- model order reduction
- ...

Essentially two uses of randomization:

(i) Preconditioning	(ii) BLAS-level
"sketch the problem"	"sketch the algorithm"

Randomized preconditioning ("sketch the problem")

Least squares problem

Given a matrix $A \in \mathbb{R}^{N \times m}$ $(N \gg m)$ and a vector $\boldsymbol{b} \in \mathbb{R}^N$, solve

$$\operatorname{argmin}_{\boldsymbol{x}} \|A\boldsymbol{x} - \boldsymbol{b}\|_2.$$

- Direct approach: column-pivoted QR. Stable. $O(Nm^2)$ ops.
- Alternative: iterative approach like LSQR producing $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$
- Convergence:

$$\|A\boldsymbol{x}_* - A\boldsymbol{x}_k\|_2 \leq 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k \|A\boldsymbol{x}_* - A\boldsymbol{x}_0\|_2$$

- $\kappa(A) \approx 1 \implies$ few iterations $\implies O(Nm)$ ops.

Randomized preconditioning (“sketch the problem”)

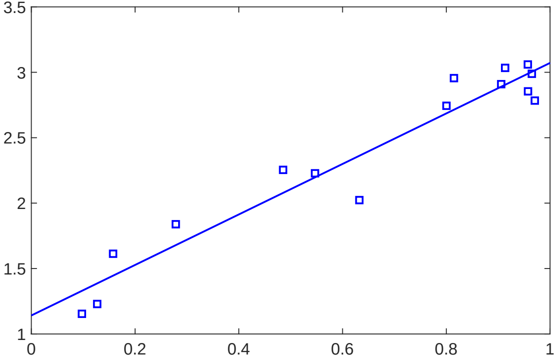
Least squares problem with preconditioner

$$\text{solve} \quad \operatorname{argmin}_{\boldsymbol{y}} \|AP\boldsymbol{y} - \boldsymbol{b}\|_2, \quad \text{then set} \quad \boldsymbol{x} = P\boldsymbol{y}$$

- Ideal preconditioner: $P = R_A^{-1}$, where $A = Q_A R_A$
- Blendenpik (Avron et al. 2010, Rokhlin-Tygert 2008):
  1. draw s-by-N random sketch matrix S with $m < s \ll N$ (e.g., $s = 2m$)
  2. compute $B = SA$
  3. compute economic QR: $B = Q_B R_B$
  4. run LSQR with preconditioner $P = R_B^{-1}$
- Can do Step 2 in $O(mN \log s)$ ops.
- $\kappa(AP) = O(1)$ with high probability!

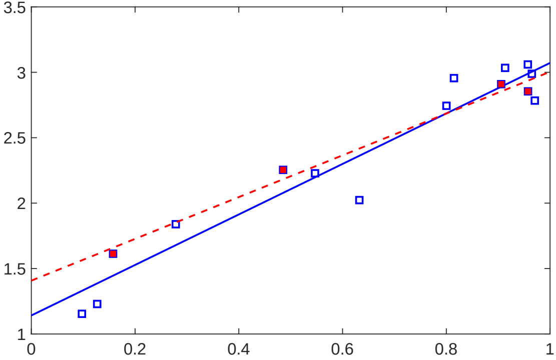
Sketch the problem: why does it work?

$$\begin{bmatrix} 1 & 0.0975 \\ 1 & 0.1270 \\ 1 & 0.1576 \\ 1 & 0.2785 \\ 1 & 0.4854 \\ 1 & 0.5469 \\ 1 & 0.6324 \\ 1 & 0.8003 \\ 1 & 0.8147 \\ 1 & 0.9058 \\ 1 & 0.9134 \\ 1 & 0.9572 \\ 1 & 0.9575 \\ 1 & 0.9649 \\ 1 & 0.9706 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \approx \begin{bmatrix} 1.1541 \\ 1.2291 \\ 1.6132 \\ 1.8388 \\ 2.2542 \\ 2.2281 \\ 2.0232 \\ 2.7440 \\ 2.9555 \\ 2.9094 \\ 3.0337 \\ 3.0597 \\ 2.8543 \\ 2.9886 \\ 2.7837 \end{bmatrix}$$



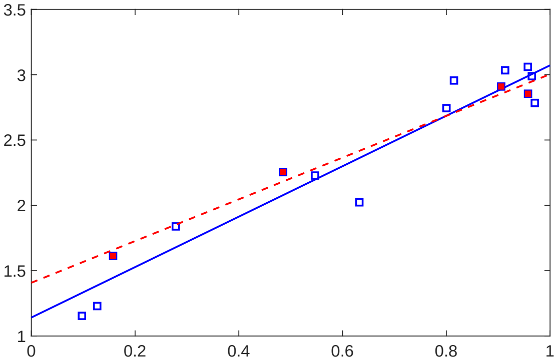
Sketch the problem: why does it work?

$$\begin{bmatrix} 1 & 0.0975 \\ 1 & 0.1270 \\ 1 & 0.1576 \\ 1 & 0.2785 \\ 1 & 0.4854 \\ 1 & 0.5469 \\ 1 & 0.6324 \\ 1 & 0.8003 \\ 1 & 0.8147 \\ 1 & 0.9058 \\ 1 & 0.9134 \\ 1 & 0.9572 \\ 1 & 0.9575 \\ 1 & 0.9649 \\ 1 & 0.9706 \end{bmatrix} \begin{bmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{bmatrix} \approx \begin{bmatrix} 1.1541 \\ 1.2291 \\ 1.6132 \\ 1.8388 \\ 2.2542 \\ 2.2281 \\ 2.0232 \\ 2.7440 \\ 2.9555 \\ 2.9094 \\ 3.0337 \\ 3.0597 \\ 2.8543 \\ 2.9886 \\ 2.7837 \end{bmatrix}$$



Sketch the problem: why does it work?

$$\begin{bmatrix} 1 & 0.0975 \\ 1 & 0.1270 \\ 1 & 0.1576 \\ 1 & 0.2785 \\ 1 & 0.4854 \\ 1 & 0.5469 \\ 1 & 0.6324 \\ 1 & 0.8003 \\ 1 & 0.8147 \\ 1 & 0.9058 \\ 1 & 0.9134 \\ 1 & 0.9572 \\ 1 & 0.9575 \\ 1 & 0.9649 \\ 1 & 0.9706 \end{bmatrix} \begin{bmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{bmatrix} \approx \begin{bmatrix} 1.1541 \\ 1.2291 \\ 1.6132 \\ 1.8388 \\ 2.2542 \\ 2.2281 \\ 2.0232 \\ 2.7440 \\ 2.9555 \\ 2.9094 \\ 3.0337 \\ 3.0597 \\ 2.8543 \\ 2.9886 \\ 2.7837 \end{bmatrix}$$



Redundancy: 15 data rows but only 2 coefficients to compute!

What about problems with no redundancy?

Sketch the algorithm

Given large sparse $A \in \mathbb{R}^{N \times N}$ and $\boldsymbol{b} \in \mathbb{R}^N$, define:

m-th order Krylov space $\quad \mathscr{K}_m(A, \boldsymbol{b}) = \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}.$

The Arnoldi process (1951) computes an orthonormal basis $V_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$:

$\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|$
For $j = 1, 2, \ldots, m$
    $\boldsymbol{w} := A\boldsymbol{v}_j$
    For $i = 1, 2, \ldots, j$
        $h_{i,j} := \langle \boldsymbol{w}, \boldsymbol{v}_i \rangle$
        $\boldsymbol{w} := \boldsymbol{w} - h_{i,j}\boldsymbol{v}_i$
    $h_{j+1,j} := \|\boldsymbol{w}\|$
    $\boldsymbol{v}_{j+1} := \boldsymbol{w}/h_{j+1,j}$

Sketch the algorithm

Given large sparse $A \in \mathbb{R}^{N\times N}$ and $\boldsymbol{b} \in \mathbb{R}^N$, define:

$$\underline{m\text{-th order Krylov space}} \quad \mathscr{K}_m(A,\boldsymbol{b}) = \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}.$$

The Arnoldi process (1951) computes an orthonormal basis $V_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$:

$\boldsymbol{v}_1 := \boldsymbol{b}/\\|\boldsymbol{b}\\|$	$O(N)$
For $j = 1, 2, \ldots, m$	
$\quad \boldsymbol{w} := A\boldsymbol{v}_j$	$O(\texttt{nnz}(A)\cdot m)$
$\quad$ For $i = 1, 2, \ldots, j$	
$\quad\quad h_{i,j} := \langle \boldsymbol{w}, \boldsymbol{v}_i\rangle$	$O(Nm^2)$
$\quad\quad \boldsymbol{w} := \boldsymbol{w} - h_{i,j}\boldsymbol{v}_i$	$O(Nm^2)$
$\quad h_{j+1,j} := \\|\boldsymbol{w}\\|$	$O(Nm)$
$\quad \boldsymbol{v}_{j+1} := \boldsymbol{w}/h_{j+1,j}$	$O(Nm)$

Subspace embedding

Assume that $S \in \mathbb{C}^{s \times N}$ is an *ε-subspace embedding* for $\mathscr{K}_m(A, \boldsymbol{b})$, $\varepsilon \in [0, 1)$.

This means that for all $\boldsymbol{u}, \boldsymbol{v} \in \mathscr{K}_m(A, \boldsymbol{b})$ we have

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle - \varepsilon \|\boldsymbol{u}\| \cdot \|\boldsymbol{v}\| \leq \langle S\boldsymbol{u}, S\boldsymbol{v} \rangle \leq \langle \boldsymbol{u}, \boldsymbol{v} \rangle + \varepsilon \|\boldsymbol{u}\| \cdot \|\boldsymbol{v}\|.$$

Equivalently, for all vectors $\boldsymbol{v} \in \mathscr{K}_m(A, \boldsymbol{b})$ we have

$$(1 - \varepsilon)\|\boldsymbol{v}\|^2 \leq \|S\boldsymbol{v}\|^2 \leq (1 + \varepsilon)\|\boldsymbol{v}\|^2.$$

In practice, S is unknown but one can construct maps that satisfy above with high probability [Sarlos 2006, Woodruff 2014, Martinsson-Tropp 2020, ...].

Usually, we choose $s \sim m/\varepsilon^2$ (e.g., $s = 2m$).

Subspace embedding: two main types

- Gaussian embedding $S \in \mathbb{R}^{s \times N}$ with entries

$$S_{i,j} = N(0, s^{-1}) \text{ i.i.d.}$$

- Subsampled Randomized Fast Transform (SRFT)

$$S = \sqrt{\frac{N}{s}}\, \Pi F D$$

  - $D \in \mathbb{R}^{N \times N}$ is a diagonal matrix of random ± 1
  - $F \in \mathbb{R}^{N \times N}$ is a fast unitary trigonometric transform (FFT, DCT2, WHT)
  - $\Pi \in \mathbb{R}^{s \times N}$ selects s random elements from a vector $\boldsymbol{v} \in \mathbb{R}^N$
  - can be applied in $O(N \log s)$ ops [Sorensen-Burrus '93, Woolfe et al. 2008]

Sketch the Arnoldi process

Idea: Replace all $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ by $\langle S\boldsymbol{u}, S\boldsymbol{v} \rangle$ (Balabanov-Nouy 2019, B.-Grigori 2022)

$\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|$	$O(N)$
For $j = 1, 2, \ldots, m$	
$\boldsymbol{w} := A\boldsymbol{v}_j$	$O(\mathtt{nnz}(A) \cdot m)$
For $i = 1, 2, \ldots, j$	
$h_{i,j} := \langle \boldsymbol{w}, \boldsymbol{v}_i \rangle$	$O(Nm^2)$
$\boldsymbol{w} := \boldsymbol{w} - h_{i,j}\boldsymbol{v}_i$	$O(Nm^2)$
$h_{j+1,j} := \|\boldsymbol{w}\|$	$O(Nm)$
$\boldsymbol{v}_{j+1} := \boldsymbol{w}/h_{j+1,j}$	$O(Nm)$

Sketch the Arnoldi process

Idea: Replace all $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ by $\langle S\boldsymbol{u}, S\boldsymbol{v} \rangle$ (Balabanov-Nouy 2019, B.-Grigori 2022)

$\boldsymbol{v}_1 := \boldsymbol{b}/\|S\boldsymbol{b}\|$	$O(N + s\log N)$
For $j = 1, 2, \ldots, m$	
$\quad \boldsymbol{w} := A\boldsymbol{v}_j$; $\widetilde{\boldsymbol{w}} := S\boldsymbol{w}$	$O(\mathtt{nnz}(A)\cdot m + s\log N)$
$\quad$ For $i = 1, 2, \ldots, j$	
$\qquad h_{i,j} := \langle \widetilde{\boldsymbol{w}}, S\boldsymbol{v}_i \rangle$	$O(sm^2)$
$\qquad \boldsymbol{w} := \boldsymbol{w} - h_{i,j}\boldsymbol{v}_i$	$O(Nm^2)$
$\quad h_{j+1,j} := \|S\boldsymbol{w}\|$	$O(Nm + s\log N)$
$\quad \boldsymbol{v}_{j+1} := \boldsymbol{w}/h_{j+1,j}$	$O(Nm)$

Sketch the Arnoldi process

Idea: Replace all $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ by $\langle S\boldsymbol{u}, S\boldsymbol{v} \rangle$ (Balabanov-Nouy 2019, B.-Grigori 2022)

$\boldsymbol{v}_1 := \boldsymbol{b}/\|S\boldsymbol{b}\|$	$O(N + s\log N)$
For $j = 1, 2, \ldots, m$	
$\quad \boldsymbol{w} := A\boldsymbol{v}_j$; $\widetilde{\boldsymbol{w}} := S\boldsymbol{w}$	$O(\mathtt{nnz}(A) \cdot m + s\log N)$
$\quad$ For $i = 1, 2, \ldots, j$	
$\qquad h_{i,j} := \langle \widetilde{\boldsymbol{w}}, S\boldsymbol{v}_i \rangle$	$O(sm^2)$
$\qquad \boldsymbol{w} := \boldsymbol{w} - h_{i,j}\boldsymbol{v}_i$	$O(Nm^2)$
$\quad h_{j+1,j} := \|S\boldsymbol{w}\|$	$O(Nm + s\log N)$
$\quad \boldsymbol{v}_{j+1} := \boldsymbol{w}/h_{j+1,j}$	$O(Nm)$

Still $O(Nm^2)$, but computes *orthonormal* SV_m and well-conditioned V_m

$$\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{1/2} \mathrm{cond}(SV_m) \leq \mathrm{cond}(V_m) \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{1/2} \mathrm{cond}(SV_m)$$

Give up on orthogonality

Idea: **Generate *some nonorthogonal* basis V_m of $\mathscr{K}_m(A, b)$ *cheaply*, then let target algorithm deal with it.** (Nakatsukasa-Tropp 2021)

Example: GMRES (Saad-Schultz 1986) computes optimal $\boldsymbol{x}_m \in \mathscr{K}_m(A, \boldsymbol{b})$ as

$$\boldsymbol{x}_m = \operatorname{argmin}_{\boldsymbol{x}} \|\boldsymbol{b} - A\boldsymbol{x}\|.$$

Let $\boldsymbol{x}_m = V_m \boldsymbol{y}_m$. Then

$$\boldsymbol{y}_m = (AV_m)^\dagger \boldsymbol{b}. \qquad O(Nm^2)$$

Use randomized sketching instead (sGMRES):

$$\widehat{\boldsymbol{y}}_m = (SAV_m)^\dagger (S\boldsymbol{b}). \qquad O(mN \log s + sm^2)$$

Applicable to other NLA problems

Randomized sketching with non-orthogonal Krylov bases also applicable to

- eigenvalue problems (Nakatsukasa-Tropp 2021)
- matrix functions (G.-Schweitzer 2023, Cortinovis-Kressner-Nakatsukasa 2023)
- matrix equations (Palitta-Schweitzer-Simoncini 2023)

Applicable to other NLA problems

Randomized sketching with non-orthogonal Krylov bases also applicable to

- eigenvalue problems (Nakatsukasa-Tropp 2021)
- matrix functions (G.-Schweitzer 2023, Cortinovis-Kressner-Nakatsukasa 2023)
- matrix equations (Palitta-Schweitzer-Simoncini 2023)

Convergence analysis of these methods complicated due to lack of orthogonality.

Partial results e.g. for $f(A)\boldsymbol{b}$ problem when $f(z)$ is Stieltjes function (like $z^{-1/2}$).

sGMRES for matrix functions

Approximate Stieltjes integral representation using sGMRES:

$$f(A)\boldsymbol{b} = \int_0^\infty (tI + A)^{-1}\boldsymbol{b}\,\mathrm{d}\mu(t) \approx V_m \int_0^\infty (tSV_m + SAV_m)^\dagger S\boldsymbol{b}\,\mathrm{d}\mu(t) =: \boldsymbol{f}_m$$

sGMRES for matrix functions

Theorem (G.-Schweitzer 2023)

*Let A be a **positive real matrix** and let $f(z)$ be a **Stieltjes function.***

Assume that S is an ε-subspace embedding for $\mathscr{K}_{m+1}(A, \boldsymbol{b})$ with $\varepsilon \in [0, 1)$.

*Then the sketched GMRES approximant $\boldsymbol{f}_m$ **converges at least linearly:***

$$\|f(A)\boldsymbol{b} - \boldsymbol{f}_m\|_{A^HA} \leq \|\boldsymbol{b}\| C_1 C_\epsilon (\sin(\beta_0))^m,$$

$C_1 = \|A\| f(\rho \|A\|^2)$, $C_\varepsilon = \sqrt{(1+\varepsilon)/(1-\varepsilon)}$, $\beta_0 = \arccos(\delta/\|A\|) \in [0, \pi/2)$.

The numbers δ and ρ depend on the numerical range of A and A^{-1}. respectively:

$$\delta := \min\left\{\Re\left(\boldsymbol{v}^H A \boldsymbol{v}\right) : \|\boldsymbol{v}\| = 1\right\}, \quad \rho := \min\left\{\Re\left(\boldsymbol{v}^H A^{-1} \boldsymbol{v}\right) : \|\boldsymbol{v}\| = 1\right\}$$

The  in the room

"Sketch-the-algorithm" Krylov approach relies on fast generation of Krylov basis.

Key challenge

Given a banded matrix $A \in \mathbb{R}^{N \times N}$ and vector $\boldsymbol{b} \in \mathbb{R}^N$. Generate well-conditioned basis V_m of span$\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}$ in $O(Nk)$ operations with fixed $k \ll m$.

Faber-Manteuffel (1984): There exists a k-term recursion

$$h_{j+1,j}\boldsymbol{v}_{j+1} = A\boldsymbol{v}_j - \sum_{i=j+2-k}^{j} h_{ij}\boldsymbol{v}_i$$

to generate *orthonormal* $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots$ if and only if $A^T = p(A)$, $p \in \mathscr{P}_{k-2}$.

Our aim

Generate *well-conditioned* V_m using k vector-vector updates per iteration, i.e.,

$$h_j \boldsymbol{v}_{j+1} = A\boldsymbol{v}_j - V_j \boldsymbol{h}_j, \quad \|\boldsymbol{h}_j\|_0 \leq k.$$

Sketch-and-select Arnoldi (G.-Simunec 2023)

At each Arnoldi iteration, select index set $I \subseteq \{1, 2, \ldots, j\}$ with $|I| \leq k$ as

$$\operatorname*{argmin}_{I} \min_{\boldsymbol{h} \in \mathbb{R}^k} \|SA\boldsymbol{v}_j - SV_j(:, I)\boldsymbol{h}\|.$$

Should be good idea to keep $\|V_j^T \boldsymbol{v}_{j+1}\|$ small as (using Demmel-Veselić 1992)

$$\mathrm{cond}([V_j, \boldsymbol{v}_{j+1}])^2 \leq \frac{1+\eta}{1-\eta}\,\mathrm{cond}(V_j)^2, \quad \eta = \sigma_{\min}(V_j)^{-2}\|V_j^T \boldsymbol{v}_{j+1}\|.$$

Solving the sparse approximation problem

The problem

$$\underset{|I|\leq k}{\operatorname{argmin}} \min_{\boldsymbol{h}\in\mathbb{R}^k} \|\boldsymbol{v} - V(\,:\,,I)\boldsymbol{h}\|.$$

is known as subset selection in statistics. It is NP-hard (Natarajan 1995).

In statistics, the residual norm $\|\boldsymbol{v} - V(\,:\,,I)\boldsymbol{h}\|$ can usually be made small.

Also arises in compressive sensing, but there the dictionary V is underdetermined.

Our setting is different

V is a basis (overdetermined) and residual norm might not reduce significantly.

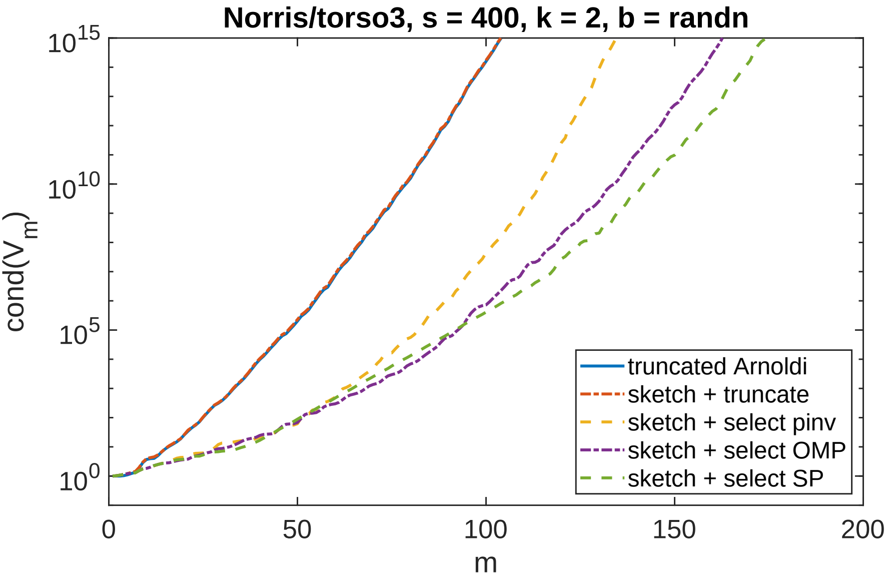
It might be necessary to develop a specific solver for this.

Some sketch-and-select methods we've tried

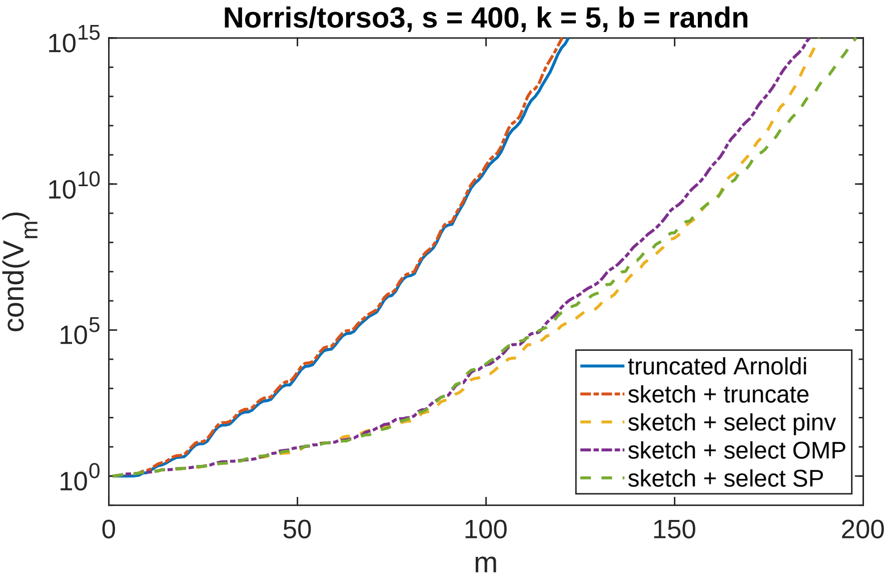
- `pinv`: compute $\boldsymbol{h} = (SV_j)^\dagger (SA\boldsymbol{v}_j)$, then keep largest k entries
- `OMP`: orthogonal matching pursuit (Pati et al. 1993) — a greedy method that selects vectors from SV_j based on correlation with the residual
- `SP`: subspace pursuit (Dai-Milenkovic 2009) — an iterative method that successively improves the index set I based on correlation with the residual

We also compare to `truncated Arnoldi`, which always projects against the last k vectors (with and without sketching).

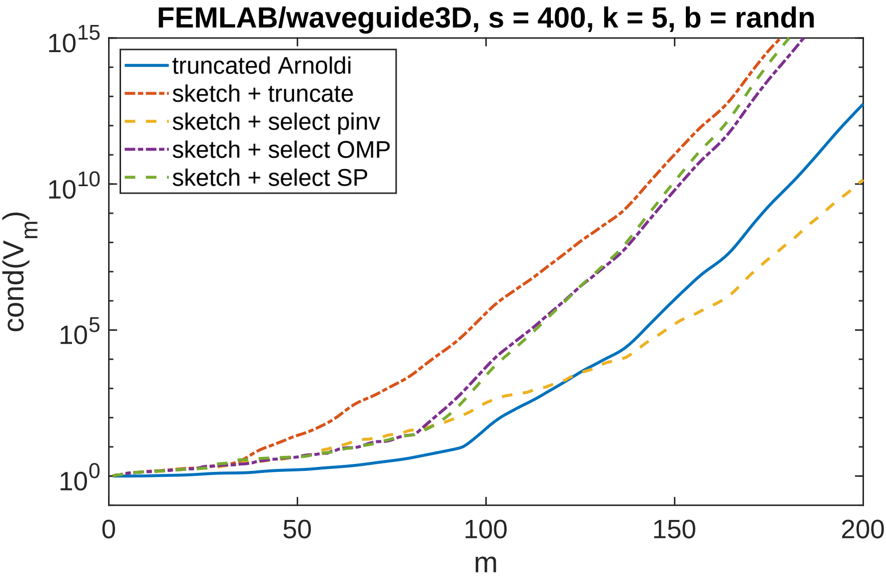
Sketch-and-select can outperform truncated Arnoldi



Sketch-and-select can outperform truncated Arnoldi



Examples where sketch + select doesn't perform well



Summary

- "Sketch the problem" vs "sketch the algorithm" paradigm
- Sketched Arnoldi inner products yields well-conditioned bases, still $O(Nm^2)$
- Use sketching to work with non-orthogonal bases (sGMRES, $f(A)\boldsymbol{b}$, ...)
- Convergence analysis available for sGMRES-type methods
- Need robust alternatives to truncated Arnoldi process
- Sketch-and-select Arnoldi promising, but needs tailored subset selection?
- Lots of new open problems to make these algorithms robust

S. Güttel & M. Schweitzer: *Randomized sketching for Krylov approximations of large-scale matrix functions*, arXiv:2208.11447, to appear in SIMAX, 2023.

S. Güttel & I. Simunec: *A sketch-and-select Arnoldi process*, arXiv:2306.03592, 2023.