Approximation of singular functions using frames

Marcus Webb

Joint with Daan Huybrechs, Vincent Coppé and Roel Matthysen

NUMA Seminar, KU Leuven
25 Oct 2018

Approximation of singular functions using frames

Marcus Webb

Joint with Daan Huybrechs, Vincent Coppé and Roel Matthysen

NUMA Seminar, KU Leuven
25 Oct 2018

Approximation of singular functions using frames

Marcus Webb

Joint with Daan Huybrechs, Vincent Coppé and Roel Matthysen

NUMA Seminar, KU Leuven
25 Oct 2018





Motivation: why singular functions?

- Some differential equations naturally have solutions with singularities:
  - Boundary layers
  - Corner singularities
  - Endpoint singularities
  - Discontinuous media: fractures/interfaces
  - Singular integral equations/fractional differential equations

Motivation: why singular functions?

- Some differential equations naturally have solutions with singularities:
  - Boundary layers
  - Corner singularities
  - Endpoint singularities
  - Discontinuous media: fractures/interfaces
  - Singular integral equations/fractional differential equations
- Location and nature of singularities known

Motivation: why singular functions?

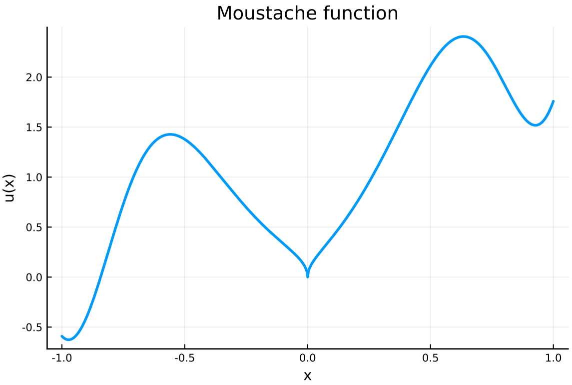
- Some differential equations naturally have solutions with singularities:
  - Boundary layers
  - Corner singularities
  - Endpoint singularities
  - Discontinuous media: fractures/interfaces
  - Singular integral equations/fractional differential equations
- Location and nature of singularities known
- Example methodologies for dealing with these:
  - Mesh refinement/domain splitting near singularity (e.g. hp-FEM)
  - Rational collocation (Weidemann 1998)
  - Enriched finite elements (Belytschko et al 1999)

Motivation: why singular functions?

- Some differential equations naturally have solutions with singularities:
  - Boundary layers
  - Corner singularities
  - Endpoint singularities
  - Discontinuous media: fractures/interfaces
  - Singular integral equations/fractional differential equations
- Location and nature of singularities known
- Example methodologies for dealing with these:
  - Mesh refinement/domain splitting near singularity (e.g. hp-FEM)
  - Rational collocation (Weidemann 1998)
  - Enriched finite elements (Belytschko et al 1999)
- Traditional spectral methods struggle in these situations.

Model problem

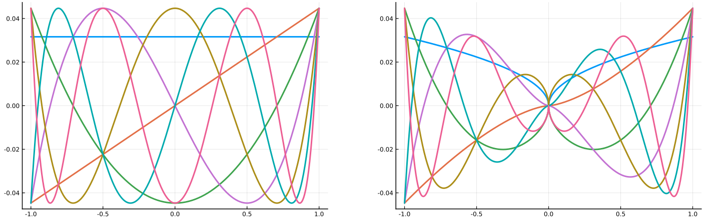
- Consider a function u of the form $u(x) = f(x) + |x|^{1/2} g(x)$
- f and g are smooth functions
- We can only sample u



- More generally, $u(x) = w_1(x) f_1(x) + \cdots + w_p(x) f_p(x)$

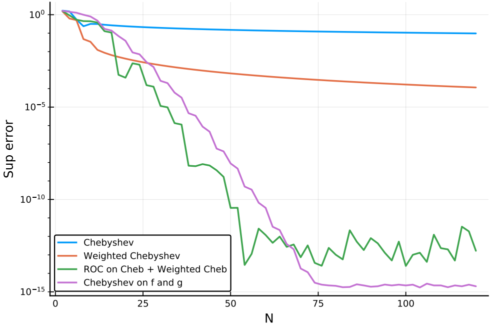
Approximating $u(x) = f(x) + |x|^{1/2}g(x)$

1. $u(x) \approx \sum_{k=0}^{N-1} a_k T_k(x)$
2. $u(x) \approx \sum_{k=0}^{N-1} b_k |x|^{1/2} T_k(x)$
3. $u(x) \approx \sum_{k=0}^{N/2-1} c_{2k} T_k(x) + c_{2k+1}|x|^{1/2} T_k(x)$



Approximating $u(x) = f(x) + |x|^{1/2} g(x)$

1. $u(x) \approx \sum_{k=0}^{N-1} a_k T_k(x)$
2. $u(x) \approx \sum_{k=0}^{N-1} b_k |x|^{1/2} T_k(x)$
3. $u(x) \approx \sum_{k=0}^{N/2-1} c_{2k} T_k(x) + c_{2k+1} |x|^{1/2} T_k(x)$



What are we computing?

- We seek the N coefficients $\mathbf{c} \in \mathbb{R}^N$ which form the least squares interpolant at $M = \gamma N$ Gauss-Chebyshev nodes $\{x_{1,M}, \ldots, x_{M,M}\}$:

$$\operatorname*{arg\,min}_{\mathbf{c} \in \mathbb{R}^N} \sum_{k=1}^{M} \left| \sum_{j=0}^{N/2-1} c_{2j} T_j(x_{k,M}) + c_{2j+1} |x_{k,M}|^{1/2} T_j(x_{k,M}) - f(x_{k,M}) \right|^2$$

What are we computing?

- We seek the N coefficients $\mathbf{c} \in \mathbb{R}^N$ which form the least squares interpolant at $M = \gamma N$ Gauss-Chebyshev nodes $\{x_{1,M}, \ldots, x_{M,M}\}$:

$$\operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{R}^N} \sum_{k=1}^{M} \left| \sum_{j=0}^{N/2-1} c_{2j} T_j(x_{k,M}) + c_{2j+1} |x_{k,M}|^{1/2} T_j(x_{k,M}) - f(x_{k,M}) \right|^2$$

- AKA **oversampled collocation**

What are we computing?

- We seek the N coefficients $\mathbf{c} \in \mathbb{R}^N$ which form the least squares interpolant at $M = \gamma N$ Gauss-Chebyshev nodes $\{x_{1,M}, \ldots, x_{M,M}\}$:

$$\underset{\mathbf{c} \in \mathbb{R}^N}{\arg\min} \sum_{k=1}^{M} \left| \sum_{j=0}^{N/2-1} c_{2j} T_j(x_{k,M}) + c_{2j+1} |x_{k,M}|^{1/2} T_j(x_{k,M}) - f(x_{k,M}) \right|^2$$

- AKA **oversampled collocation**
- Equivalent to the least squares solution of the tall, skinny linear system, $A\mathbf{c} = \mathbf{b}$, where $A \in \mathbb{R}^{M \times N}$,

$$A_{k,2j} = T_j(x_{k,M}), \qquad A_{k,2j+1} = |x_{k,M}|^{1/2} T_j(x_{k,M}),$$

$$b_k = f(x_{k,M}), \qquad k = 1, 2, \ldots, M, j = 0, 1, \ldots, N/2 - 1.$$

FrameFun vs ApproxFun (Chebfun)

- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients.

FrameFun vs ApproxFun (Chebfun)

- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients. Just like Chebfun!
- A and A^* can be applied to a vector fast using a DCT.

FrameFun vs ApproxFun (Chebfun)

- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients. Just like Chebfun!
- A and A^* can be applied to a vector fast using a DCT. Just like Chebfun!

FrameFun vs ApproxFun (Chebfun)

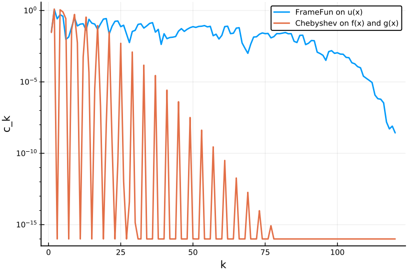
- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients. Just like Chebfun!
- A and A^* can be applied to a vector fast using a DCT. Just like Chebfun!
- A is exponentially ill-conditioned ($\kappa(A) \geq 4^N$). Sad!

FrameFun vs ApproxFun (Chebfun)

- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients. Just like Chebfun!
- A and A^* can be applied to a vector fast using a DCT. Just like Chebfun!
- A is exponentially ill-conditioned ($\kappa(A) \geq 4^N$). Sad!?

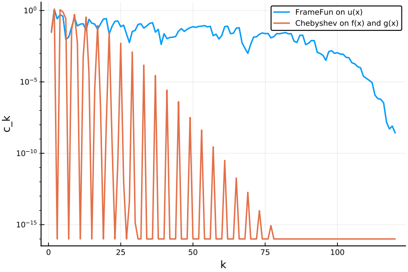
FrameFun vs ApproxFun (Chebfun)

- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients. Just like Chebfun!
- A and A^* can be applied to a vector fast using a DCT. Just like Chebfun!
- A is exponentially ill-conditioned $(\kappa(A) \geq 4^N)$. Sad!?
- Coefficient size and error correlation:



FrameFun vs ApproxFun (Chebfun)

- The collocation matrix A is a **transform** between **coefficients** $\mathbf{c}$ and **values** of the function with those coefficients. Just like Chebfun!
- A and A^* can be applied to a vector fast using a DCT. Just like Chebfun!
- A is exponentially ill-conditioned $(\kappa(A) \geq 4^N)$. Sad!?
- Coefficient size and error correlation:



Adaptivity and frames: Coppé-Huybrechs (in prep.)

Solving ill-conditioned systems

- Ill-conditioned least squares problem: $A \in \mathbb{R}^{M \times N}$, $\mathbf{b} \in \mathbb{R}^M$ $(M = \gamma N)$,

$$\arg\min_{\mathbf{x} \in \mathbb{R}^N} \|A\mathbf{x} - b\|_2^2$$

Solving ill-conditioned systems

- Ill-conditioned least squares problem: $A \in \mathbb{R}^{M \times N}$, $\mathbf{b} \in \mathbb{R}^M$ $(M = \gamma N)$,

$$\arg\min_{\mathbf{x} \in \mathbb{R}^N} \|A\mathbf{x} - b\|_2^2$$

- Many solutions with small residual.

Solving ill-conditioned systems

- Ill-conditioned least squares problem: $A \in \mathbb{R}^{M \times N}$, $\mathbf{b} \in \mathbb{R}^M$ $(M = \gamma N)$,
$$\arg\min_{\mathbf{x} \in \mathbb{R}^N} \|A\mathbf{x} - b\|_2^2$$
- Many solutions with small residual. The ε-**regularised** SVD solution is,

$$A = U \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & \sigma_N \end{bmatrix} V^*, \quad \mathbf{x} = V \begin{bmatrix} \sigma_1^{-1} & & & & \\ & \ddots & & & \\ & & \sigma_r^{-1} & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} U^* b$$

where $\sigma_k < \varepsilon \iff k > r$.

Solving ill-conditioned systems

- Ill-conditioned least squares problem: $A \in \mathbb{R}^{M\times N}$, $\mathbf{b} \in \mathbb{R}^M$ ($M = \gamma N$),
$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} \|A\mathbf{x} - b\|_2^2$$
- Many solutions with small residual. The ε-**regularised** SVD solution is,
$$A = U \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & \sigma_N \end{bmatrix} V^*, \quad \mathbf{x} = V \begin{bmatrix} \sigma_1^{-1} & & & & \\ & \ddots & & & \\ & & \sigma_r^{-1} & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} U^* b$$
where $\sigma_k < \varepsilon \iff k > r$.
- Backslash computes the ε-regularised pivoted-QR solution

Solving ill-conditioned systems

- Ill-conditioned least squares problem: $A \in \mathbb{R}^{M\times N}$, $\mathbf{b} \in \mathbb{R}^M$ $(M = \gamma N)$,
$$\arg\min_{\mathbf{x}\in\mathbb{R}^N} \|A\mathbf{x} - b\|_2^2$$
- Many solutions with small residual. The ε-**regularised** SVD solution is,

$$A = U \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & \sigma_N \end{bmatrix} V^*, \quad \mathbf{x} = V \begin{bmatrix} \sigma_1^{-1} & & & & \\ & \ddots & & & \\ & & \sigma_r^{-1} & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} U^* b$$

where $\sigma_k < \varepsilon \iff k > r$.
- Backslash computes the ε-regularised pivoted-QR solution
- $O(N^3)$ flops – **very slow!**

Adcock-Huybrechs Theorems

Oversampled Collocation Theorem (Adcock–Huybrechs 2018)

Let $\Phi = \{\varphi_k\}_{k=1}^{\infty}$ be a **frame** for $L^2(\Omega)$ and let $\{w_{k,M} f(x_{k,M})\}_{k=1}^{M}$ be "good" samples for any $f \in L^2(\Omega)$.
If the entries of $A \in \mathbb{R}^{M \times N}$ are $a_{k,j} = w_{k,M}\varphi_j(x_{k,M})$, and $b_k = w_{k,M} u(x_{k,M})$, then the ε-regularised solution $u^{\varepsilon,M,N}(x)$ satisfies

$$\|u^{\varepsilon,M,N} - u\|_{L^2(\Omega)} \leq C^{\varepsilon}_{M,N} \left(\left\| \sum_{k=1}^{N} v_k \varphi_k - u \right\|_{L^2(\Omega)} + \varepsilon \|\mathbf{v}\|_2 \right),$$

for any $\mathbf{v} \in \ell^2$, where $\sup_N \limsup_{M \to \infty} C^{\varepsilon}_{M,N} \leq C < \infty$.

Adcock-Huybrechs Theorems

Oversampled Collocation Theorem (Adcock–Huybrechs 2018)

Let $\Phi = \{\varphi_k\}_{k=1}^{\infty}$ be a **frame** for $L^2(\Omega)$ and let $\{w_{k,M} f(x_{k,M})\}_{k=1}^{M}$ be "good" samples for any $f \in L^2(\Omega)$.
If the entries of $A \in \mathbb{R}^{M\times N}$ are $a_{k,j} = w_{k,M}\varphi_j(x_{k,M})$, and $b_k = w_{k,M} u(x_{k,M})$, then the ε-regularised solution $u^{\varepsilon,M,N}(x)$ satisfies

$$\|u^{\varepsilon,M,N} - u\|_{L^2(\Omega)} \leq C^{\varepsilon}_{M,N} \left(\left\| \sum_{k=1}^{N} v_k \varphi_k - u \right\|_{L^2(\Omega)} + \varepsilon \|\mathbf{v}\|_2 \right),$$

for any $\mathbf{v} \in \ell^2$, where $\sup_N \limsup_{M\to\infty} C^{\varepsilon}_{M,N} \leq C < \infty$.

Furthermore, the RHS converges to $\mathcal{O}(\varepsilon)$ as $N \to \infty$, with **sufficient oversampling** M.

What is a frame?

- A **frame** is a set of functions $\Phi = \{\varphi_k\}_{k=1}^{\infty} \subset \mathcal{H}$ (inner product space) such that

$$f \mapsto \|\left(\langle \varphi_k, f \rangle\right)_{k=1}^{\infty}\|_{\ell^2} \quad \text{and} \quad f \mapsto \|f\|_{\mathcal{H}}$$

are **equivalent norms** on $\mathcal{H}$.

What is a frame?

- A **frame** is a set of functions $\Phi = \{\varphi_k\}_{k=1}^{\infty} \subset \mathcal{H}$ (inner product space) such that

$$f \mapsto \| \left(\langle \varphi_k, f \rangle \right)_{k=1}^{\infty} \|_{\ell^2} \quad \text{and} \quad f \mapsto \|f\|_{\mathcal{H}}$$

  are **equivalent norms** on $\mathcal{H}$.
- The set $\varphi_{2k} = T_k$, $\varphi_{2k+1} = w \cdot T_k$, satisfies

$$\|f\|^2 \inf_{x \in [-1,1]} |1+|w(x)|^2| \leq \sum_{k=0}^{\infty} |\langle \varphi_k, f \rangle|^2 \leq \|f\|^2 \sup_{x \in [-1,1]} |1+|w(x)|^2|$$

What is a frame?

- A **frame** is a set of functions $\Phi = \{\varphi_k\}_{k=1}^{\infty} \subset \mathcal{H}$ (inner product space) such that

$$f \mapsto \| \left(\langle \varphi_k, f \rangle\right)_{k=1}^{\infty} \|_{\ell^2} \quad \text{and} \quad f \mapsto \|f\|_{\mathcal{H}}$$

  are **equivalent norms** on $\mathcal{H}$.

- The set $\varphi_{2k} = T_k$, $\varphi_{2k+1} = w \cdot T_k$, satisfies

$$\|f\|^2 \inf_{x \in [-1,1]} |1+|w(x)|^2| \leq \sum_{k=0}^{\infty} |\langle \varphi_k, f \rangle|^2 \leq \|f\|^2 \sup_{x \in [-1,1]} |1+|w(x)|^2|$$

- We have a frame if $1 + |w(x)|^2$ is **bounded above** and **below**

Dual frames

- Typical focus: dual frame or "inversion of the frame operator"
- A dual frame $\tilde{\Phi} = \{\tilde{\varphi}_k\}_{k=1}^{\infty}$ satisfies

$$f = \sum_{k=1}^{\infty} \langle \tilde{\varphi}_k, f \rangle \varphi_k$$

Dual frames

- Typical focus: dual frame or "inversion of the frame operator"
- A dual frame $\tilde{\Phi} = \{\tilde{\varphi}_k\}_{k=1}^{\infty}$ satisfies

$$f = \sum_{k=1}^{\infty} \langle \tilde{\varphi}_k, f \rangle \varphi_k$$

$$\Phi = \{T_k(x)\}_{k=1}^{\infty} \cup \{w(x)T_k(x)\}_{k=1}^{\infty}$$

$$\tilde{\Phi} = \left\{ \frac{T_k(x)}{1 + |w(x)|^2} \right\}_{k=1}^{\infty} \cup \left\{ \frac{w(x)T_k(x)}{1 + |w(x)|^2} \right\}_{k=1}^{\infty}$$

- These coefficients, $c_k = \langle \tilde{\varphi}_k, f \rangle$, converge too slowly! ROC gives better approximations.

Aside: Solving a low-rank system fast

Aside: Solving a low-rank system fast

- Let $A \in \mathbb{R}^{M \times r}$, where $r \ll M$. Then the SVD of A can be computed and inverted in $\mathcal{O}(Mr^2)$ operations.

Aside: Solving a low-rank system fast

- Let $A \in \mathbb{R}^{M \times r}$, where $r \ll M$. Then the SVD of A can be computed and inverted in $\mathcal{O}(Mr^2)$ operations.
- What about if $A \in \mathbb{R}^{M \times N}$ has rank r?

Aside: Solving a low-rank system fast

- Let $A \in \mathbb{R}^{M \times r}$, where $r \ll M$. Then the SVD of A can be computed and inverted in $\mathcal{O}(Mr^2)$ operations.
- What about if $A \in \mathbb{R}^{M \times N}$ has rank r?

Randomised least squares solver for $A\mathbf{x} = \mathbf{b}$

1. $W = \mathrm{randn}(N, r + 20)$
2. Least squares solve for $\mathbf{y} \in \mathbb{R}^{r+20}$: $(AW)\mathbf{y} = \mathbf{b}$
3. $\mathbf{x} = W\mathbf{y} \in \mathbb{R}^N$

Aside: Solving a low-rank system fast

- Let $A \in \mathbb{R}^{M \times r}$, where $r \ll M$. Then the SVD of A can be computed and inverted in $\mathcal{O}(Mr^2)$ operations.
- What about if $A \in \mathbb{R}^{M \times N}$ has rank r?

Randomised least squares solver for $A\mathbf{x} = \mathbf{b}$

1. $W = \mathrm{randn}(N, r + 20)$
2. Least squares solve for $\mathbf{y} \in \mathbb{R}^{r+20}$: $(AW)\mathbf{y} = \mathbf{b}$
3. $\mathbf{x} = W\mathbf{y} \in \mathbb{R}^N$

Theorem (Using techniques in Halko, Martinsson, Tropp 2011)

The computed solution $\mathbf{x}$ satisfies,

$$\|A\mathbf{x} - b\|_2 \leq \|A\mathbf{v} - b\|_2 + \kappa_{r,N} \cdot \left(\sum_{k>r} \sigma_k^2\right)^{1/2} \cdot \|\mathbf{v}\|_2, \qquad \forall \mathbf{v} \in \mathbb{R}^N,$$

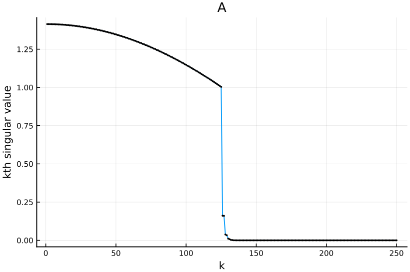
where $\kappa_{r,N}$ is a random variable such that
$\mathbb{P}\left[\kappa_{r,N} > 16 + 5\sqrt{r}\right] < 2.89 \times 10^{-9}$.

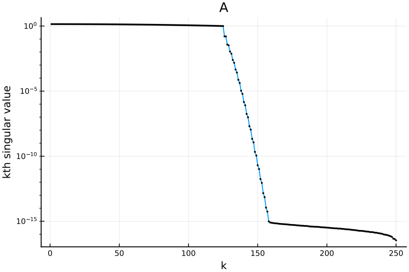
The plunge region

- Let $A \in \mathbb{R}^{M \times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated frame, $\{T_k\}_{k=0}^{N/2-1} \cup \{|x|^{1/2} T_k(x)\}_{k=0}^{N/2-1}$.

The plunge region

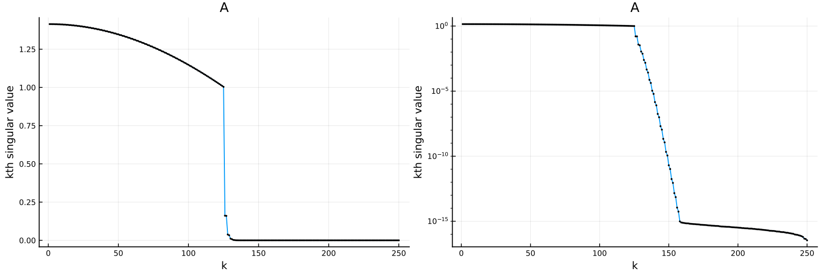
- Let $A \in \mathbb{R}^{M \times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated frame, $\{T_k\}_{k=0}^{N/2-1} \cup \{|x|^{1/2} T_k(x)\}_{k=0}^{N/2-1}$.





The plunge region

- Let $A \in \mathbb{R}^{M\times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated frame, $\{T_k\}_{k=0}^{N/2-1} \cup \{|x|^{1/2}T_k(x)\}_{k=0}^{N/2-1}$.



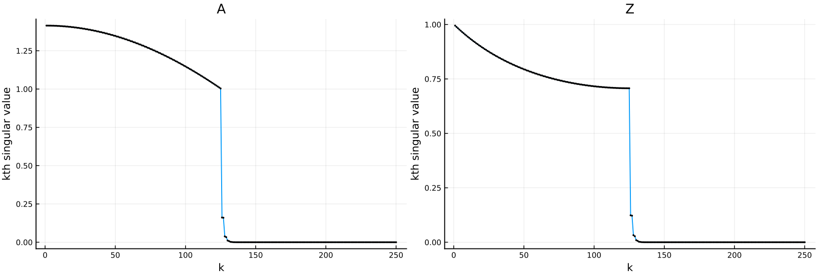
- For weighted sums of trigonometric bases, the number of singular values in $(\varepsilon, 1-\varepsilon)$ is $\mathcal{O}(\log(N))$ (see Adcock-Huybrechs FNA paper and Webb (in prep.)).
- The big-$\mathcal{O}$ depends on ε and the BV norms of the weights. Precise dependence is an **open problem**.

Dual frame isolates plunge region

Let $A \in \mathbb{R}^{M\times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated frame, $\{T_k\}_{k=0}^{N/2-1} \cup \{|x|^{1/2}T_k(x)\}_{k=0}^{N/2-1}$.
Let $Z \in \mathbb{R}^{M\times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated **dual frame**,
$\{T_k(x)/(1+|x|)\}_{k=0}^{N/2-1} \cup \{|x|^{1/2}T_k(x)/(1+|x|)\}_{k=0}^{N/2-1}$.

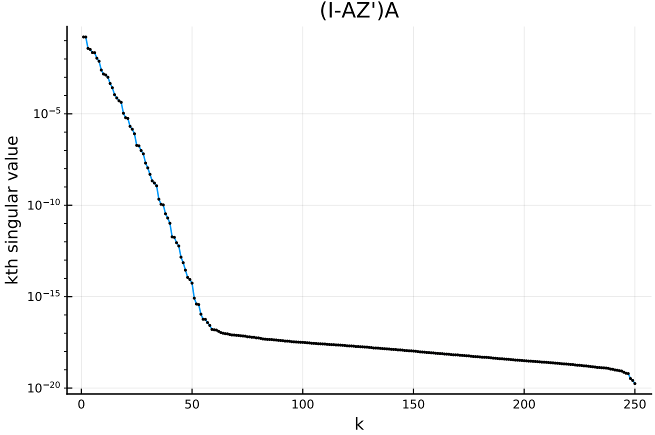
Dual frame isolates plunge region

Let $A \in \mathbb{R}^{M\times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated frame, $\{T_k\}_{k=0}^{N/2-1} \cup \{|x|^{1/2}T_k(x)\}_{k=0}^{N/2-1}$.
Let $Z \in \mathbb{R}^{M\times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated **dual frame**,
$\{T_k(x)/(1+|x|)\}_{k=0}^{N/2-1} \cup \{|x|^{1/2}T_k(x)/(1+|x|)\}_{k=0}^{N/2-1}$.



Dual frame isolates plunge region

Let $A \in \mathbb{R}^{M \times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated frame, $\{T_k\}_{k=0}^{N/2-1} \cup \{|x|^{1/2} T_k(x)\}_{k=0}^{N/2-1}$.
Let $Z \in \mathbb{R}^{M \times N}$ be the collocation matrix in M Gauss-Chebyshev points for the N-truncated **dual frame**,
$\{T_k(x)/(1+|x|)\}_{k=0}^{N/2-1} \cup \{|x|^{1/2} T_k(x)/(1+|x|)\}_{k=0}^{N/2-1}$.



The AZ algorithm - $A, Z \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$

AZ Algorithm for a least squares solution to $A\mathbf{x} = b$:

1. Solve $(I - AZ^*)A\mathbf{x}_1 = (I - AZ^*)b$
2. $\mathbf{x}_2 = Z^*(b - A\mathbf{x}_1)$
3. $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$

The AZ algorithm - $A, Z \in \mathbb{R}^{M\times N}$, $b \in \mathbb{R}^M$

AZ Algorithm for a least squares solution to $A\mathbf{x} = b$:

1. Solve $(I - AZ^*)A\mathbf{x}_1 = (I - AZ^*)b$
2. $\mathbf{x}_2 = Z^*(b - A\mathbf{x}_1)$
3. $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$

- Residual:
$$\begin{aligned} b - A\mathbf{x} &= b - A\mathbf{x}_1 - A\mathbf{x}_2 \\ &= b - A\mathbf{x}_1 - AZ^*(b - A\mathbf{x}_1) \\ &= (I - AZ^*)(b - A\mathbf{x}_1). \end{aligned}$$

The AZ algorithm - $A, Z \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$

AZ Algorithm for a least squares solution to $A\mathbf{x} = b$:

1. Solve $(I - AZ^*)A\mathbf{x}_1 = (I - AZ^*)b$
2. $\mathbf{x}_2 = Z^*(b - A\mathbf{x}_1)$
3. $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$

- Residual:
$$\begin{aligned} b - A\mathbf{x} &= b - A\mathbf{x}_1 - A\mathbf{x}_2 \\ &= b - A\mathbf{x}_1 - AZ^*(b - A\mathbf{x}_1) \\ &= (I - AZ^*)(b - A\mathbf{x}_1). \end{aligned}$$
- If $\mathrm{rank}_\varepsilon((I - AZ^*)A) = \mathrm{rk}_N$, and $A\mathbf{v}$, $Z^*\mathbf{w}$ require mul_N operations, then, in total,

$$\mathcal{O}(\mathrm{mul}_N \cdot \mathrm{rk}_N + N \cdot \mathrm{rk}_N^2) \text{ operations.}$$

The AZ algorithm - $A, Z \in \mathbb{R}^{M\times N}$, $b \in \mathbb{R}^M$

AZ Algorithm for a least squares solution to $A\mathbf{x} = b$:

1. Solve $(I - AZ^*)A\mathbf{x}_1 = (I - AZ^*)b$
2. $\mathbf{x}_2 = Z^*(b - A\mathbf{x}_1)$
3. $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$

- Residual:
$$\begin{aligned} b - A\mathbf{x} &= b - A\mathbf{x}_1 - A\mathbf{x}_2 \\ &= b - A\mathbf{x}_1 - AZ^*(b - A\mathbf{x}_1) \\ &= (I - AZ^*)(b - A\mathbf{x}_1). \end{aligned}$$
- If $\mathrm{rank}_\varepsilon((I - AZ^*)A) = \mathrm{rk}_N$, and $A\mathbf{v}$, $Z^*\mathbf{w}$ require mul_N operations, then, in total,
$$\mathcal{O}(\mathrm{mul}_N \cdot \mathrm{rk}_N + N \cdot \mathrm{rk}_N^2) \text{ operations.}$$
- Our model problem: $\mathcal{O}(N \log^2(N))$

The AZ algorithm - $A, Z \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$

AZ Algorithm for a least squares solution to $A\mathbf{x} = b$:

1. Solve $(I - AZ^*)A\mathbf{x}_1 = (I - AZ^*)b$
2. $\mathbf{x}_2 = Z^*(b - A\mathbf{x}_1)$
3. $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$

- Residual:
$$\begin{aligned} b - A\mathbf{x} &= b - A\mathbf{x}_1 - A\mathbf{x}_2 \\ &= b - A\mathbf{x}_1 - AZ^*(b - A\mathbf{x}_1) \\ &= (I - AZ^*)(b - A\mathbf{x}_1). \end{aligned}$$
- If $\mathrm{rank}_\varepsilon((I - AZ^*)A) = \mathrm{rk}_N$, and $A\mathbf{v}$, $Z^*\mathbf{w}$ require mul_N operations, then, in total,
$$\mathcal{O}(\mathrm{mul}_N \cdot \mathrm{rk}_N + N \cdot \mathrm{rk}_N^2) \text{ operations.}$$
- Our model problem: $\mathcal{O}(N \log^2(N))$
- See Coppé-Huybrechs-Matthysen-Webb (in prep.)

Discussion

Effective algorithms:

- Adcock-Huybrechs: for frames use regularised oversampled collocation
- Coefficients and adaptivity don't behave like in ApproxFun/Chebfun

Fast algorithms:

- Plunge region
- Fast randomised linear algebra
- The AZ algorithm
- Implemented in Julia package FrameFun

Discussion

Effective algorithms:

- Adcock-Huybrechs: for frames use regularised oversampled collocation
- Coefficients and adaptivity don't behave like in ApproxFun/Chebfun

Fast algorithms:

- Plunge region
- Fast randomised linear algebra
- The AZ algorithm
- Implemented in Julia package FrameFun

Several papers in prep.!