Local low-rank approximation for the high-frequency Helmholtz equation

Steffen Börm

Christian-Albrechts-Universität zu Kiel

SIAM ALA 2018, Hong Kong

Overview

Overview

Helmholtz equation

Goal: Solve the Helmholtz equation

$$\begin{aligned} \Delta u(x) + \kappa^2 u(x) &= 0 && \text{for all } x \in \Omega \subseteq \mathbb{R}^3, \\ u(x) &= f(x) && \text{for all } x \in \partial\Omega. \end{aligned}$$

Approach: Boundary integral formulation

$$u(x) = \int_{\partial\Omega} g(x,y) \frac{\partial u}{\partial n}(y)\, dy - \int_{\partial\Omega} \frac{\partial g}{\partial n(y)}(x,y) u(y)\, dy \quad \text{for all } x \in \Omega$$

with fundamental solution

$$g(x,y) = \frac{\exp(\iota\kappa\|x-y\|)}{4\pi\|x-y\|}.$$

Challenge: Standard discretization leads to dense matrix.

Fast summation

Standard approach: Approximate g by degenerate kernel,

$$g(x,y) \approx \sum_{\nu=1}^{k} v_\nu(x) \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)}$$

Fast summation

Standard approach: Approximate g by degenerate kernel,

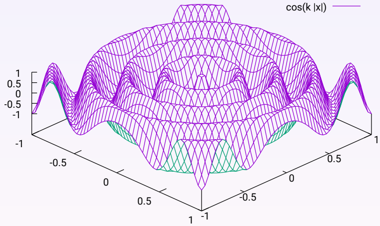
$$\int_{\partial\Omega} g(x,y)u(y)\,dy \approx \sum_{\nu=1}^{k} v_\nu(x) \int_{\partial\Omega} \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)} u(y)\,dy.$$

Fast summation

Standard approach: Approximate g by degenerate kernel,

$$\int_{\partial\Omega} g(x,y)u(y)\,dy \approx \sum_{\nu=1}^{k} v_\nu(x) \int_{\partial\Omega} \sum_{\mu=1}^{k} s_{\nu\mu} \overline{w_\mu(y)} u(y)\,dy.$$

Challenge: Helmholtz kernel oscillates rapidly if κ is large.



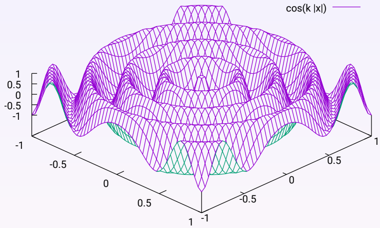
$$g(x,y) = \frac{\exp(\iota\kappa\|x-y\|)}{\|x-y\|}.$$

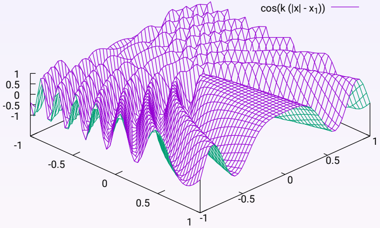
Consequence: Standard approximation schemes require high rank k.

Directional approximation

Idea: Divide spherical wave by plane wave in direction c, $\|c\| = 1$:

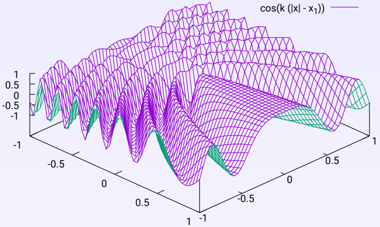
$$g(x, y) = \exp(\iota\kappa\langle x - y, c\rangle) \underbrace{\frac{\exp(\iota\kappa(\|x - y\| - \langle x - y, c\rangle))}{4\pi\|x - y\|}}_{=g_c(x,y)}.$$





✎ Brandt (1991), Engquist/Ying (2007), Messner/Schanz/Darve (2012), Bebendorf/Kuske/Venn (2015)

Directional smoothness



Admissibility conditions: Let $B_t, B_s \subseteq \mathbb{R}^3$ be axis-parallel boxes with centers m_t, m_s. g_c is smooth in $B_t \times B_s$ if

$$\kappa \left\| \frac{m_t - m_s}{\|m_t - m_s\|} - c \right\| \lesssim \frac{1}{\max\{\operatorname{diam}(B_t), \operatorname{diam}(B_s)\}},$$

$$\kappa \max\{\operatorname{diam}^2(B_t), \operatorname{diam}^2(B_s)\} \lesssim \operatorname{dist}(B_t, B_s),$$

$$\max\{\operatorname{diam}(B_t), \operatorname{diam}(B_s)\} \lesssim \operatorname{dist}(B_t, B_s).$$

Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$g_c(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y),$$

$$g(x,y) = \exp(\iota\kappa\langle x-y, c\rangle) g_c(x,y)$$

Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$g_c(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y),$$

$$g(x,y) = \exp(\iota\kappa\langle x-y, c\rangle) g_c(x,y)$$

$$\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \underbrace{\exp(\langle x, c\rangle)\mathcal{L}_{t,\nu}(x)}_{=:v_{tc,\nu}(x)} \underbrace{g_c(\xi_{t,\nu}, \xi_{s,\mu})}_{=:s_{ts,\nu\mu}} \underbrace{\overline{\exp(\langle y, c\rangle)\mathcal{L}_{s,\mu}(y)}}_{=:\overline{v_{sc,\mu}(y)}}$$

Directional interpolation

Idea: Choose admissible boxes $B_t, B_s \subseteq \mathbb{R}^3$ and interpolate g_c.

$$g_c(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \mathcal{L}_{s,\mu}(y),$$

$$g(x,y) = \exp(\iota\kappa\langle x-y, c\rangle) g_c(x,y)$$

$$\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \underbrace{\exp(\langle x, c\rangle)\mathcal{L}_{t,\nu}(x)}_{=:v_{tc,\nu}(x)} \underbrace{g_c(\xi_{t,\nu}, \xi_{s,\mu})}_{=:s_{ts,\nu\mu}} \underbrace{\overline{\exp(\langle y, c\rangle)\mathcal{L}_{s,\mu}(y)}}_{=:\overline{v_{sc,\mu}(y)}}$$

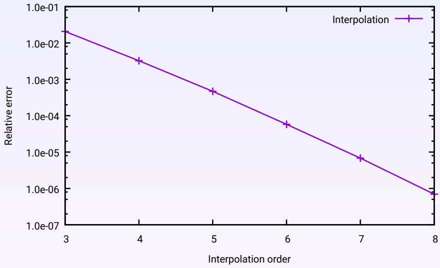
$\mathcal{DH}^2$-matrix: Low-rank approximation of admissible submatrices

$$G|_{t\times s} \approx V_{tc} S_{ts} V_{sc}^*$$

with multilevel representation of basis matrices (V_{tc}).

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles, $\kappa h \approx 0.6$.

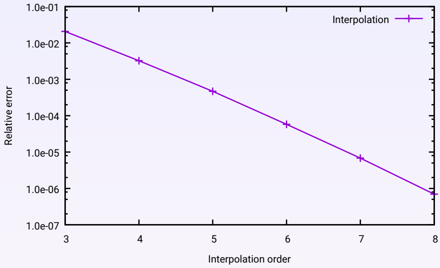


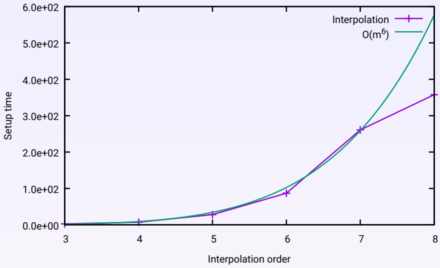
Observations:

- Exponential convergence.

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles, $\kappa h \approx 0.6$.



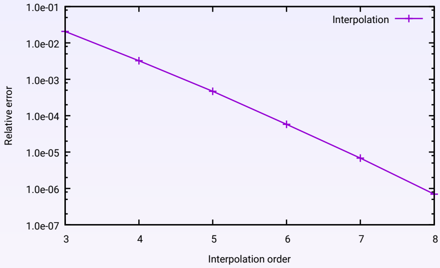


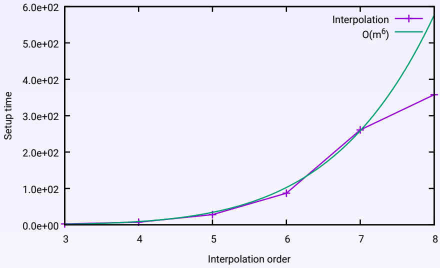
Observations:

- Exponential convergence.
- Setup time proportional to k^2, in the range of a few minutes.

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles, $\kappa h \approx 0.6$.





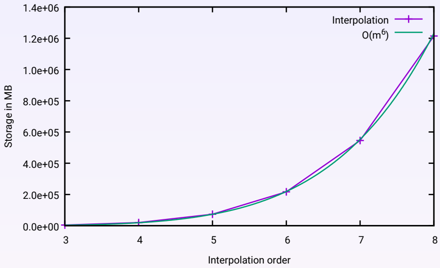
Observations:

- Exponential convergence.
- Setup time proportional to k^2, in the range of a few minutes.

Messner/Schanz/Darve (2012), B./Melenk (2017)

Directional interpolation: Storage

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32\,768$ triangles.

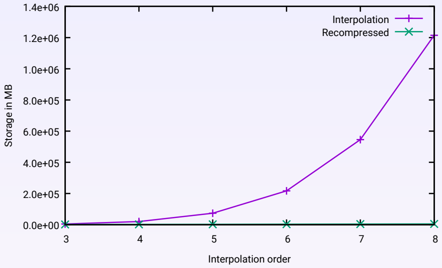


Observation: 1.2 TB for 32 768 degrees of freedom is inacceptable.

Overview

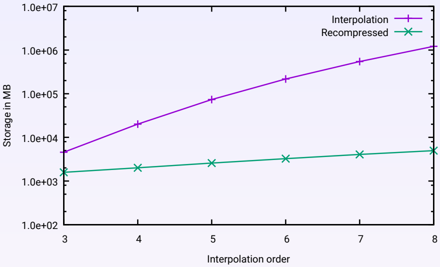
Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.



Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.

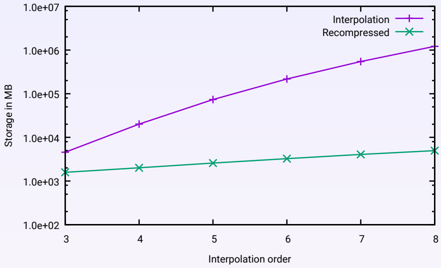


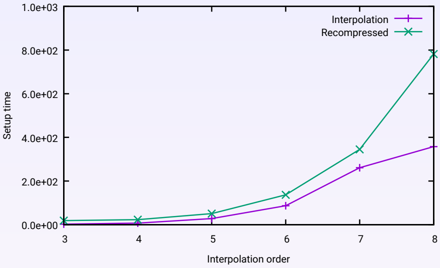
Observations:

- Storage reduced by more than a factor of more than 100.

Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.



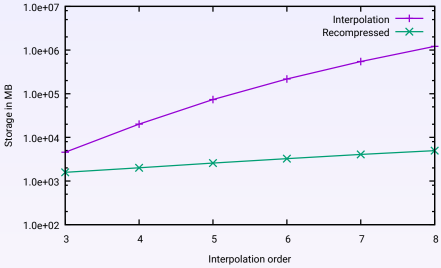


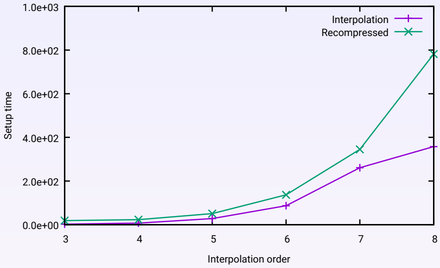
Observations:

- Storage reduced by more than a factor of more than 100.
- Setup time approximately doubled.

Algebraic recompression

Experiment: Apply algebraic recompression to the approximation constructed by directional interpolation.





Observations:

- Storage reduced by more than a factor of more than 100.
- Setup time approximately doubled.

✎ B. (2017), B./Börst (in preparation)

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$\begin{pmatrix} V_{tc} S_{ts_1} V_{s_1 c}^* & \cdots & V_{tc} S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}$$

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$V_{tc} \underbrace{\begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}}_{=: B_{tc}^*}.$$

The matrix B_{tc} has only k columns. $\rightarrow$ Low-rank factorization.

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$V_{tc} \underbrace{\begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}}_{=:B_{tc}^*}.$$

The matrix B_{tc} has only k columns. $\rightarrow$ Low-rank factorization.

Compression: Construct isometric matrix Q_{tc} of low rank such that

$$Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*,$$

i.e., by SVD or rank-revealing QR.

Uniform approximation

Goal: Replace V_{tc} with a basis Q_{tc} of lower rank.

Since V_{tc} is required by multiple blocks, we have to approximate

$$V_{tc} \underbrace{\begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}}_{=:B_{tc}^*}.$$

The matrix B_{tc} has only k columns. $\rightarrow$ Low-rank factorization.

Compression: Construct isometric matrix Q_{tc} of low rank such that

$$Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*,$$

i.e., by SVD or rank-revealing QR.

Challenge: B_{tc} has a large number of rows.

Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{tc} = P_{tc} R_{tc}$, $R_{tc} \in \mathbb{C}^{k \times k}$.

$$
\begin{aligned}
B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\
\to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}.
\end{aligned}
$$

Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{tc} = P_{tc} R_{tc}$, $R_{tc} \in \mathbb{C}^{k \times k}$.

$$\begin{aligned} B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\ \to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}. \end{aligned}$$

Total weights: Construct thin QR factorization $\widehat{B}_{tc} = \widehat{P}_{tc} Z_{tc}$.

Compression: Find low-rank Q_{tc} with $Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* \approx V_{tc} Z_{tc}^*$.

Basis weights

Goal: $Q_{tc} Q_{tc}^* V_{tc} B_{tc}^* \approx V_{tc} B_{tc}^*$.

Idea: Applying orthogonal transformations from the right does not change the approximation quality.

Basis weights: Construct QR factorizations $V_{tc} = P_{tc} R_{tc}$, $R_{tc} \in \mathbb{C}^{k \times k}$.

$$
\begin{aligned}
B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix} \\
\to \widehat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}.
\end{aligned}
$$

Total weights: Construct thin QR factorization $\widehat{B}_{tc} = \widehat{P}_{tc} Z_{tc}$.

Compression: Find low-rank Q_{tc} with $Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* \approx V_{tc} Z_{tc}^*$.

Error control: Replace S_{ts} by $S_{ts} / \| R_{tc} S_{ts} R_{sc}^* \|$ and ensure $\| Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* - V_{tc} Z_{tc}^* \| \leq \epsilon$ to control block-wise relative error.

Overview

Cross approximation

Problem: For high accuracies, the coupling matrices S_{ts} are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.
$\to$ Construction of total weights Z_{tc} fairly slow.

Idea: Since S_{ts} corresponds to the smooth kernel function g_c, we may use cross approximation (essentially rank-revealing LR) to obtain

$$S_{ts} \approx C_{ts} D_{ts}^*,$$

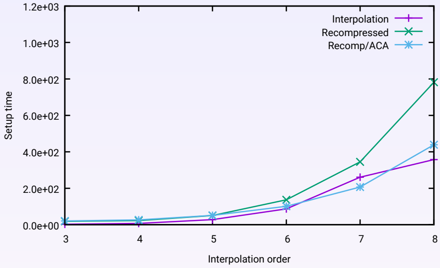
where C_{ts}, D_{ts} have a reduced number $\hat{k} \leq k$ of columns.

Condensation: Again apply isometric transformations from the right: QR factorization yields $D_{ts} = P_{ts} R_{ts}$ with $R_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$ and P_{ts} isometric.

$$S_{ts} \approx C_{ts} D_{ts}^* = C_{ts} R_{ts}^* P_{ts}^* = \widehat{C}_{ts} P_{ts}^*, \qquad \widehat{C} \in \mathbb{C}^{k \times \hat{k}}.$$

Algebraic compression with cross approximation

Experiment: Algebraic recompression of directional interpolation, coupling matrices approximated by cross approximation.

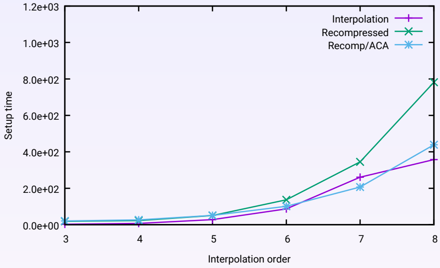


Observations:

- Storage reduction and accuracy as before.

Algebraic compression with cross approximation

Experiment: Algebraic recompression of directional interpolation, coupling matrices approximated by cross approximation.



Observations:

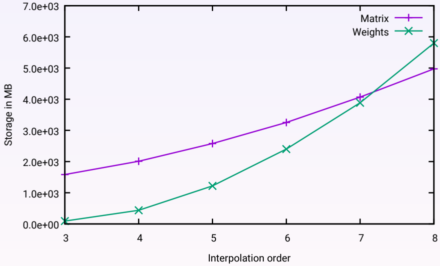
- Storage reduction and accuracy as before.
- Setup time only moderately increased compared to original.

Basis weights revisited

Problem: In order to compute the total weights Z_{tc}, we need a QR factorization of the matrix B_{tc},

$$B_{tc}^* = \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_\ell} V_{s_\ell c}^* \end{pmatrix}.$$

The first step is to replace V_{sc} by a QR factorization $V_{sc} = P_{sc} R_{sc}$.
$\to$ We have to store R_{sc} for all clusters and directions.



Compressed basis weights

Idea: Consider products $S_{ts} V_{sc}^*$, exploit low-rank property of S_{ts}.

Approach: Find isometric P_{sc} of low rank with

$$P_{sc} P_{sc}^* V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix} \approx V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix}.$$

With $R_{sc} := P_{sc}^* V_{sc}$ we have $S_{ts} V_{sc}^* \approx S_{ts} R_{sc}^* P_{sc}^*$ for all submatrices.

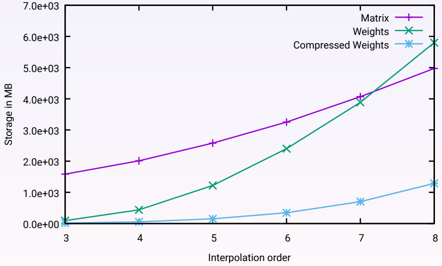
Compressed basis weights

Idea: Consider products $S_{ts} V_{sc}^*$, exploit low-rank property of S_{ts}.

Approach: Find isometric P_{sc} of low rank with

$$P_{sc} P_{sc}^* V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix} \approx V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix}.$$

With $R_{sc} := P_{sc}^* V_{sc}$ we have $S_{ts} V_{sc}^* \approx S_{ts} R_{sc}^* P_{sc}^*$ for all submatrices.



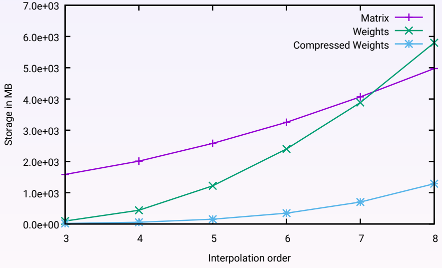
Compressed basis weights

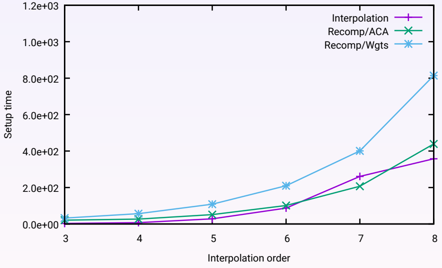
Idea: Consider products $S_{ts} V_{sc}^*$, exploit low-rank property of S_{ts}.

Approach: Find isometric P_{sc} of low rank with

$$P_{sc} P_{sc}^* V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix} \approx V_{sc} \begin{pmatrix} S_{t_1 s}^* & \cdots & S_{t_\ell s}^* \end{pmatrix}.$$

With $R_{sc} := P_{sc}^* V_{sc}$ we have $S_{ts} V_{sc}^* \approx S_{ts} R_{sc}^* P_{sc}^*$ for all submatrices.





Summary

Interpolation is fast and reliable, but needs too much storage.

Recompression based on rank-revealing factorizations can significantly reduce the storage requirements.

Modifications like cross approximation and compressed weights improve performance, particularly for high accuracies.