Hybrid compression of boundary element matrices for high-frequency Helmholtz problems

Steffen Börm
with C. Börst and J. M. Melenk

Christian-Albrechts-Universität zu Kiel

Zurich Colloquium in Applied and Computational Mathematics
Overview

1. Introduction
2. $DH^2$-matrices
3. Compression
4. Modifications
5. Summary
Overview

1. Introduction

2. $\mathcal{D}H^2$-matrices

3. Compression

4. Modifications

5. Summary
Helmholtz equation

Goal: Solve the Helmholtz equation

\[ \Delta u(x) + \kappa^2 u(x) = 0 \quad \text{for all } x \in \Omega \subseteq \mathbb{R}^3, \]
\[ u(x) = f(x) \quad \text{for all } x \in \partial \Omega. \]

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

\[ g(x, y) = \frac{\exp(i \kappa \| x - y \|)}{4\pi \| x - y \|}. \]
Discretization

Galerkin’s method with finite-element bases \((\varphi_i)_{i=1}^n\) and \((\psi_j)_{j=1}^n\) leads to matrix \(G \in \mathbb{C}^{n \times n}\) with entries

\[
g_{ij} = \int_{\partial \Omega} \varphi_i(x) \int_{\partial \Omega} g(x, y) \psi_j(y) \, dy \, dx, \quad \text{for all } i, j \in [1 : n].
\]

Challenges:

- \(G\) is not sparse.

We are interested in the high-frequency case \(\kappa \sim n\), so the kernel function \(g\) oscillates rapidly.
Discretization

Galerkin’s method with finite-element bases \((\varphi_i)_{i=1}^n\) and \((\psi_j)_{j=1}^n\) leads to matrix \(G \in \mathbb{C}^{n \times n}\) with entries

\[
g_{ij} = \int_{\partial\Omega} \varphi_i(x) \int_{\partial\Omega} g(x, y) \overline{\psi_j(y)} \, dy \, dx, \quad \text{for all } i, j \in [1 : n].
\]

Challenges:

- \(G\) is not sparse.
- We are interested in the high-frequency case \(\kappa^2 \sim n\), so the kernel function \(g\) oscillates rapidly.
Overview

1. Introduction
2. $DH^2$-matrices
3. Compression
4. Modifications
5. Summary
Low-rank approximation

**Standard approach:** Approximate $g$ by degenerate kernel,

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

Discretization yields rank-$k$ matrix factorization

$G \approx VSW^*$

$V_i \nu = \int_{\partial \Omega} \phi_i(x) v_{\nu}(x) \, dx$

$W_j \mu = \int_{\partial \Omega} \psi_j(y) w_{\mu}(y) \, dy$

**Challenges:**

Degenerate functions converge only locally.

In standard methods, oscillations lead to large rank-\( k \).
Low-rank approximation

Standard approach: Approximate $g$ by degenerate kernel,

$$ g(x, y) \approx \sum_{\nu=1}^{k} v_{\nu}(x) \sum_{\mu=1}^{k} s_{\nu \mu} w_{\mu}(y) = \sum_{\nu, \mu=1}^{k} s_{\nu \mu} v_{\nu} \otimes w_{\mu}(x, y). $$
Low-rank approximation

Standard approach: Approximate $g$ by degenerate kernel,

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

Discretization yields rank-$k$ matrix factorization

$$G \approx VSW^*, \quad V_{i\nu} = \int_{\partial \Omega} \varphi_i(x) v_{\nu}(x) \, dx, \quad W_{j\mu} = \int_{\partial \Omega} \psi_j(y) w_{\mu}(y) \, dy.$$
Low-rank approximation

Standard approach: Approximate $g$ by degenerate kernel,

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

Discretization yields rank-$k$ matrix factorization

$$G \approx VSW^*, \quad V_{i\nu} = \int_{\partial \Omega} \varphi_i(x) v_{\nu}(x) \, dx, \quad W_{j\mu} = \int_{\partial \Omega} \psi_j(y) w_{\mu}(y) \, dy.$$ 

Challenges:
- Degenerate functions converge only locally.
- In standard methods, oscillations lead to large rank $k$. 
Directional approximation

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

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

Admissibility conditions

Local approximation: Since approximation works only locally, we have to identify suitable subsets of \( \mathbb{R}^3 \times \mathbb{R}^3 \).

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

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

\( \kappa \)
Admissibility conditions

Local approximation: Since approximation works only locally, we have to identify suitable subsets of $\mathbb{R}^3 \times \mathbb{R}^3$.

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

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

Local approximation: Since approximation works only locally, we have to identify suitable subsets of \( \mathbb{R}^3 \times \mathbb{R}^3 \).

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

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

\[
\kappa \max\{\text{diam}^2(B_t), \text{diam}^2(B_s)\} \lesssim \text{dist}(B_t, B_s),
\]
Admissibility conditions

Local approximation: Since approximation works only locally, we have to identify suitable subsets of \( \mathbb{R}^3 \times \mathbb{R}^3 \).

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

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

\[
\kappa \max\{\text{diam}^2(B_t), \text{diam}^2(B_s)\} \lesssim \text{dist}(B_t, B_s),
\]

\[
\max\{\text{diam}(B_t), \text{diam}(B_s)\} \lesssim \text{dist}(B_t, B_s).
\]
Directional interpolation

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

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

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

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

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

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

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

$$= : v_{tc,\nu}(x)$$

$$= : s_{ts,\nu,\mu}$$

$$= : v_{sc,\mu}(y)$$
Directional interpolation

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

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

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

$$\approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \exp(\iota \kappa \langle x, c \rangle) \mathcal{L}_{t,\nu}(x) g_c(\xi_{t,\nu}, \xi_{s,\mu}) \frac{\iota \kappa \exp(\langle y, c \rangle) \mathcal{L}_{s,\mu}(y)}{=: v_{tc,\nu}(x)}$$

Algebraic counterpart: For subsets $t := \{ i \in [1 : n] : \text{supp } \varphi_i \subseteq B_t \}$ and $s := \{ j \in [1 : n] : \text{supp } \psi_j \subseteq B_s \}$, we obtain

$$G|_{t \times s} \approx V_{t c} S_{t s} V_{s c}^*.$$
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

Idea: Recursively split index set and matrix to find admissible submatrices.
Tree structures

**Idea:** Recursively split index set and matrix to find admissible submatrices.

- **Cluster tree** describes subdivision of the index set $\mathcal{I} = [1 : n]$ into localized subsets $t, s \subseteq \mathcal{I}$ with bounding boxes $B_t, B_s \subseteq \mathbb{R}^3$.
- **Block tree** described corresponding subdivision of the matrix into admissible submatrices $G|_{t \times s}$ and a remainder.
Nested directional bases

Problem: To satisfy the first admissibility condition, we require $\sim \kappa^2 \text{diam}^2(B_t)$ directions for each cluster $t$.

$\rightarrow$ Storing $(V_{tc})$ directly requires $O(nk\kappa^2 \log n)$ units of storage.

$\rightarrow O(n^2k \log n)$ for high frequencies, $\kappa^2 \sim n$. 
Nested directional bases

Problem: To satisfy the first admissibility condition, we require \( \sim \kappa^2 \text{diam}^2(B_t) \) directions for each cluster \( t \).

\[ \Rightarrow \text{Storing } (V_{tc}) \text{ directly requires } \mathcal{O}(nk\kappa^2 \log n) \text{ units of storage.} \]

\[ \Rightarrow \mathcal{O}(n^2k \log n) \text{ for high frequencies, } \kappa^2 \sim n. \]

Idea: Express \( V_{tc} \) in terms of \( V_{t'c'} \) for sons \( t' \) of \( t \).

\[ v_{tc,\nu}(x) = \exp(\iota \kappa \langle x, c \rangle) \mathcal{L}_{t,\nu}(x) \]
Nested directional bases

Problem: To satisfy the first admissibility condition, we require
\( \sim \kappa^2 \text{diam}^2(B_t) \) directions for each cluster \( t \).

\( \rightarrow \) Storing \( (V_{tc}) \) directly requires \( \mathcal{O}(nk\kappa^2 \log n) \) units of storage.

\( \rightarrow \mathcal{O}(n^2k \log n) \) for high frequencies, \( \kappa^2 \sim n \).

Idea: Express \( V_{tc} \) in terms of \( V_{t'c'} \) for sons \( t' \) of \( t \).

\[ v_{tc,\nu}(x) = \exp(\nu \kappa \langle x, c' \rangle) \exp(\nu \kappa \langle x, c - c' \rangle) L_{t,\nu}(x) \]
Nested directional bases

Problem: To satisfy the first admissibility condition, we require 
\(~ \kappa^2 \text{diam}^2(B_t)\) directions for each cluster \(t\).

\(\rightarrow\) Storing \((V_{tc})\) directly requires \(O(nk\kappa^2 \log n)\) units of storage.

\(\rightarrow\) \(O(n^2 k \log n)\) for high frequencies, \(\kappa^2 \sim n\).

Idea: Express \(V_{tc}\) in terms of \(V_{t'c'}\) for sons \(t'\) of \(t\).

\[v_{tc,\nu}(x) = \exp(\nu \kappa \langle x, c' \rangle) \exp(\nu \kappa \langle x, c - c' \rangle) L_{t,\nu}(x)\]

\[ \approx \exp(\nu \kappa \langle x, c' \rangle) \sum_{\nu'=1}^{k} \exp(\nu \kappa \langle \xi_{t'},\nu', c - c' \rangle) L_{t,\nu}(\xi_{t'},\nu') L_{t',\nu'}(x)\]
Nested directional bases

Problem: To satisfy the first admissibility condition, we require
\( \sim \kappa^2 \operatorname{diam}^2(B_t) \) directions for each cluster \( t \).
→ Storing \( (V_{tc}) \) directly requires \( O(nk\kappa^2 \log n) \) units of storage.
→ \( O(n^2k \log n) \) for high frequencies, \( \kappa^2 \sim n \).

Idea: Express \( V_{tc} \) in terms of \( V_{t'c'} \) for sons \( t' \) of \( t \).

\[
v_{tc,\nu}(x) = \exp(\iota\kappa \langle x, c' \rangle) \exp(\iota\kappa \langle x, c - c' \rangle) \mathcal{L}_{t,\nu}(x) \\
\approx \exp(\iota\kappa \langle x, c' \rangle) \sum_{\nu' = 1}^{k} \exp(\iota\kappa \langle \xi_{t'},\nu', c - c' \rangle) \mathcal{L}_{t,\nu}(\xi_{t'},\nu') \mathcal{L}_{t',\nu'}(x) \\
= \sum_{\nu' = 1}^{k} e_{t'c,\nu',\nu} v_{t'c',\nu'}(x)
\]

Approach: Store only \( k \times k \) transfer matrices \( E_{tc'} \) for non-leaf clusters.
Result: Complexity

$O(\kappa^2 n^2 \log n)$, $O(\kappa^2 n^2)$ in the high-frequency case with $\kappa^2 \approx n$.
Complexity

Result: Complexity $O(nk + \kappa^2 k^2 \log n)$, $O(nk^2 \log n)$ in the high-frequency case with $\kappa^2 \sim n$.
Result: Complexity $O(nk + \kappa^2 k^2 \log n)$, $O(nk^2 \log n)$ in the high-frequency case with $\kappa^2 \sim n$

Directional interpolation: Convergence

Experiment: Compress Helmholtz Galerkin matrix corresponding to the unit sphere approximated by with $n = 32768$ 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.

Directional interpolation: Storage

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

![Graph showing storage vs. interpolation order](image)

**Observation:** 1.2 TB for 32,768 degrees of freedom is unacceptable.
Overview

1 Introduction

2 $DH^2$-matrices

3 Compression

4 Modifications

5 Summary
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.

S. Börm (CAU Kiel)
Helmholtz low-rank approximation
10th of October, 2018
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.

舵/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

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

The matrix $B_{tc}$ has only $k$ columns.

→ 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.
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

$$V_{tc} \left( S_{ts_1} V_{s_1 c}^* \cdots S_{ts_\ell} V_{s_\ell c}^* \right).$$

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

\[
V_{tc} \left( S_{ts_1} V_{s_1c}^* \cdots S_{ts_{\ell}} V_{s_{\ell}c}^* \right) := B_{tc}^*.
\]

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

$$V_{tc} \begin{pmatrix} S_{ts_1} & V_{s_1c}^* & \cdots & S_{ts_\ell} & V_{s_\ell c}^* \end{pmatrix} =: B_{tc}^*.$$

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 \( V_{sc} = P_{sc} R_{sc} \), \( R_{sc} \in \mathbb{C}^{k \times k} \).

\[
B_{tc}^* = \begin{pmatrix} S_{ts_1} V_{s_1 c} & \cdots & S_{ts_{\ell}} V_{s_{\ell} c} \end{pmatrix}
\]
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 \( V_{sc} = P_{sc} R_{sc} \), \( R_{sc} \in \mathbb{C}^{k \times k} \).

\[
\begin{align*}
B_{tc}^* &= \begin{pmatrix} S_{ts_1} V_{s_1 c}^* & \cdots & S_{ts_{\ell}} V_{s_{\ell} c}^* \end{pmatrix} \\
\rightarrow \quad \hat{B}_{tc}^* &= \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_{\ell}} R_{s_{\ell} c}^* \end{pmatrix}.
\end{align*}
\]
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 $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.  

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

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

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 $V_{sc} = P_{sc} R_{sc}$, $R_{sc} \in \mathbb{C}^{k \times k}$.

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

$$\rightarrow \hat{B}_{tc}^* = \begin{pmatrix} S_{ts_1} R_{s_1 c}^* & \cdots & S_{ts_\ell} R_{s_\ell c}^* \end{pmatrix}.$$ 

Total weights: Construct thin QR factorization $\hat{B}_{tc} = \hat{P}_{tc} Z_{tc}$.
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 \( V_{sc} = P_{sc} R_{sc} \), \( R_{sc} \in \mathbb{C}^{k \times k} \).

\[
B^{*}_{tc} = \begin{pmatrix} S_{ts1} V^{*}_{s1c} & \cdots & S_{ts\ell} V^{*}_{s\ell c} \end{pmatrix} \\
\rightarrow \hat{B}^{*}_{tc} = \begin{pmatrix} S_{ts1} R^{*}_{s1c} & \cdots & S_{ts\ell} R^{*}_{s\ell c} \end{pmatrix}
\]

Total weights: Construct thin QR factorization \( \hat{B}_{tc} = \hat{P}_{tc} Z_{tc} \).

Compression: Find low-rank \( Q_{tc} \) with \( Q_{tc} Q^{*}_{tc} V_{tc} Z^{*}_{tc} \approx V_{tc} Z^{*}_{tc} \), e.g., by singular value decomposition.

Error control achieved by weighting the matrices \( S_{ts} \).
Nested bases

Goal: Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix}$
Nested bases

**Goal:** Obtain nested bases, i.e., transfer matrices $F_{t^i_c}$ with

$$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix} = \begin{pmatrix} Q_{t_1 c_1} \\ Q_{t_2 c_2} \end{pmatrix} \begin{pmatrix} F_{t_1 c} \\ F_{t_2 c} \end{pmatrix} = U_{tc} \hat{Q}_{tc}$$

Compression now takes the form

$$Q_{tc} Q_{tc}^* = V_{tc} Z_{tc}^* \approx V_{tc} Z_{tc}^*,$$

with the significantly smaller matrix

$$\hat{V}_{tc} := U_{tc} V_{tc} \in \mathbb{C}^{(2 \tilde{k}) \times k}.$$
Nested bases

**Goal:** Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

$$Q_{tc} = \begin{pmatrix} Q_{t_1c_1} F_{t_1c} \\ Q_{t_2c_2} F_{t_2c} \end{pmatrix} = \begin{pmatrix} Q_{t_1c_1} \\ Q_{t_2c_2} \end{pmatrix} \begin{pmatrix} F_{t_1c} \\ F_{t_2c} \end{pmatrix} = U_{tc} \hat{Q}_{tc}$$

Compression now takes the form

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

$$\hat{Q}_{tc} \hat{Q}_{tc}^* U_{tc}^* V_{tc} Z_{tc}^* \approx U_{tc}^* V_{tc} Z_{tc}^* ,$$

$$\hat{Q}_{tc} \hat{Q}_{tc}^* \hat{V}_{tc} Z_{tc}^* \approx \hat{V}_{tc} Z_{tc}^*$$

with the significantly smaller matrix $\hat{V}_{tc} := U_{tc}^* V_{tc} \in \mathbb{C}^{(2\tilde{k}) \times k}$. 
Nested bases

Goal: Obtain nested bases, i.e., transfer matrices $F_{t'c}$ with

$$Q_{tc} = \begin{pmatrix} Q_{t_1 c_1} F_{t_1 c} \\ Q_{t_2 c_2} F_{t_2 c} \end{pmatrix} = \begin{pmatrix} Q_{t_1 c_1} \\ Q_{t_2 c_2} \end{pmatrix} \begin{pmatrix} F_{t_1 c} \\ F_{t_2 c} \end{pmatrix} = U_{tc} \hat{Q}_{tc}$$

Compression now takes the form

$$Q_{tc} Q_{tc}^* V_{tc} Z_{tc}^* \approx V_{tc} Z_{tc}^*,$$
$$\hat{Q}_{tc} \hat{Q}_{tc}^* U_{tc}^* V_{tc} Z_{tc}^* \approx U_{tc}^* V_{tc} Z_{tc}^*,$$
$$\hat{Q}_{tc} \hat{Q}_{tc}^* \hat{V}_{tc} Z_{tc}^* \approx \hat{V}_{tc} Z_{tc}^*$$

with the significantly smaller matrix $\hat{V}_{tc} := U_{tc}^* V_{tc} \in \mathbb{C}^{(2\tilde{k}) \times k}$.

Result: Algebraic recompression in $O(nk + \kappa^2 k^2 \log n)$ operations, $O(nk^2 \log n)$ in high-frequency case.
Overview

1. Introduction
2. $\mathcal{DH}^2$-matrices
3. Compression
4. Modifications
5. Summary
Cross approximation

**Problem:** For high accuracies, the coupling matrices $S_{ts}$ are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.

→ 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 to obtain

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

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

Problem: For high accuracies, the coupling matrices $S_{ts}$ are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.

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 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 $R_{sc} D_{ts} = P_{ts} \hat{R}_{ts}$ with $\hat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}, P_{ts}$ isometric.

$$S_{ts} R_{sc}^*$$
Cross approximation

Problem: For high accuracies, the coupling matrices $S_{ts}$ are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.

→ 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 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 $R_{sc} D_{ts} = P_{ts} \hat{R}_{ts}$ with $\hat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, $P_{ts}$ isometric.

$$S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^*$$
Cross approximation

Problem: For high accuracies, the coupling matrices $S_{ts}$ are large, e.g., $m = 8$ leads to dimension $8^3 = 512$. → 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 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 $R_{sc} D_{ts} = P_{ts} \hat{R}_{ts}$ with $\hat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, $P_{ts}$ isometric.

$$S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^* = C_{ts} (R_{sc} D_{ts})^*$$
Cross approximation

Problem: For high accuracies, the coupling matrices $S_{ts}$ are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.

$\rightarrow$ 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 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 $R_{sc} D_{ts} = P_{ts} \hat{R}_{ts}$ with $\hat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}, P_{ts}$ isometric.

$$ S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^* = C_{ts} (R_{sc} D_{ts})^* = C_{ts} \hat{R}_{ts}^* P_{ts}^*. $$

Cross approximation

**Problem:** For high accuracies, the coupling matrices $S_{ts}$ are large, e.g., $m = 8$ leads to dimension $8^3 = 512$.

$\rightarrow$ 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 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 $R_{sc} D_{ts} = P_{ts} \hat{R}_{ts}$ with $\hat{R}_{ts} \in \mathbb{C}^{\hat{k} \times \hat{k}}$, $P_{ts}$ isometric.

$$S_{ts} R_{sc}^* \approx C_{ts} D_{ts}^* R_{sc}^* = C_{ts} (R_{sc} D_{ts})^* = C_{ts} \hat{R}_{ts} P_{ts}^*.$$

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 $B_{tc},$

$$B_{tc}^* = \left( S_{ts_1} V_{s_1}^* \cdots S_{ts\ell} V_{s_\ell}^* \right).$$

The first step is to replace $V_{sc}$ by a QR factorization $V_{sc} = P_{sc} R_{sc}$.

$\rightarrow$ 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

$$P_{sc} P_{sc}^* V_{sc} (S_{t_1 s}^* \cdots S_{t_\ell s}^*) \approx V_{sc} (S_{t_1 s}^* \cdots S_{t_\ell s}^*) .$$

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

$$P_{sc} P_{sc}^* V_{sc} \left( S_{t_1s}^* \cdots S_{t_\ell s}^* \right) \approx V_{sc} \left( S_{t_1s}^* \cdots S_{t_\ell s}^* \right).$$

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

$$P_{sc} P_{sc}^* V_{sc} \left( S_{t_1s}^* \cdots S_{t_\ell s}^* \right) \approx V_{sc} \left( S_{t_1s}^* \cdots S_{t_\ell s}^* \right).$$

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

Work in progress: Generalization to complex wave numbers $\kappa = a + b \iota$. Take advantage of exponential decay via modified admissibility.

Experiment: $\kappa = 16 + 16\iota$, ACA and compressed weights.
Work in progress: Generalization to complex wave numbers $\kappa = a + b\nu$. Take advantage of exponential decay via modified admissibility.

Experiment: $\kappa = 16 + 16\nu$, ACA and compressed weights.
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.