$\mathcal{H}^2$-Matrix Preconditioners for Elliptic Problems

Steffen Börm
(with Knut Reimer and Dirk Boysen)

Christian-Albrechts-Universität zu Kiel

EMG 2012, Schwetzingen, 13th of August, 2012

Preconditioners for elliptic PDEs

Model problem:

$$-\operatorname{div} \quad \operatorname{grad} u \;(x) = f(x) \qquad \text{for all } x \in \Omega,$$
$$u(x) = g(x) \qquad \text{for all } x \in \partial\Omega$$

with two-dimensional domain Ω.

Preconditioners for elliptic PDEs

Model problem:

$$-\operatorname{div} \quad \operatorname{grad} u \ (x) = f(x) \qquad \text{for all } x \in \Omega,$$
$$u(x) = g(x) \qquad \text{for all } x \in \partial\Omega$$

with two-dimensional domain Ω.

Desirable properties of a good preconditioner:

- Fast convergence.

Preconditioners for elliptic PDEs

Model problem:

$$-\operatorname{div}(E \operatorname{grad} u)(x) = f(x) \qquad \text{for all } x \in \Omega,$$
$$u(x) = g(x) \qquad \text{for all } x \in \partial\Omega$$

with two-dimensional domain Ω.

Desirable properties of a good preconditioner:

- Fast convergence.
- Robust with respect to non-smooth coefficients.

Preconditioners for elliptic PDEs

Model problem:

$$-\operatorname{div}(E \operatorname{grad} u)(x) = f(x) \qquad \text{for all } x \in \Omega,$$
$$u(x) = g(x) \qquad \text{for all } x \in \partial\Omega$$

with three- or two-dimensional domain Ω.

Desirable properties of a good preconditioner:

- Fast convergence.
- Robust with respect to non-smooth coefficients.
- Only moderately dependent on spatial dimension.

Preconditioners for elliptic PDEs

Model problem:

$$-\operatorname{div}(E \operatorname{grad} u)(x) = f(x) \qquad \text{for all } x \in \Omega,$$
$$u(x) = g(x) \qquad \text{for all } x \in \partial\Omega$$

with three- or two-dimensional domain Ω.

Desirable properties of a good preconditioner:

- Fast convergence.
- Robust with respect to non-smooth coefficients.
- Only moderately dependent on spatial dimension.

Proposal: $\mathcal{H}$- or $\mathcal{H}^2$-matrix preconditioners.

Overview

Interior regularity

Theory: Given a subdomain $\omega \subseteq \Omega$ with $\operatorname{dist}(\omega, \partial\Omega) > 0$, we have

$$\|u\|_{H^{m+1}(\omega)} \lesssim \frac{1}{\operatorname{dist}(\omega, \partial\Omega)^m} (\|f\|_{H^{m-1}(\Omega)} + \|u\|_{L^2(\Omega)})$$

if f and E are sufficiently smooth in Ω.

Interior regularity

Theory: Given a subdomain $\omega \subseteq \Omega$ with $\operatorname{dist}(\omega, \partial\Omega) > 0$, we have

$$\|u\|_{H^{m+1}(\omega)} \lesssim \frac{1}{\operatorname{dist}(\omega, \partial\Omega)^m} (\|f\|_{H^{m-1}(\Omega)} + \|u\|_{L^2(\Omega)})$$

if f and E are sufficiently smooth in Ω.

Idea: Approximate $u|_\omega$ by a polynomial $\tilde{u} \in \Pi_m$ to obtain

$$\|u - \tilde{u}\|_{L^2(\omega)} \lesssim \left(\frac{\operatorname{diam}(\omega)}{\operatorname{dist}(\omega, \partial\Omega)} \right)^m (\|f\|_{H^{m-1}(\Omega)} + \|u\|_{L^2(\Omega)}).$$

Interior regularity

Theory: Given a subdomain $\omega \subseteq \Omega$ with $\operatorname{dist}(\omega, \partial\Omega) > 0$, we have

$$\|u\|_{H^{m+1}(\omega)} \lesssim \frac{1}{\operatorname{dist}(\omega, \partial\Omega)^m} (\|f\|_{H^{m-1}(\Omega)} + \|u\|_{L^2(\Omega)})$$

if f and E are sufficiently smooth in Ω.

Idea: Approximate $u|_\omega$ by a polynomial $\tilde{u} \in \Pi_m$ to obtain

$$\|u - \tilde{u}\|_{L^2(\omega)} \lesssim \left(\frac{\operatorname{diam}(\omega)}{\operatorname{dist}(\omega, \partial\Omega)} \right)^m (\|f\|_{H^{m-1}(\Omega)} + \|u\|_{L^2(\Omega)}).$$

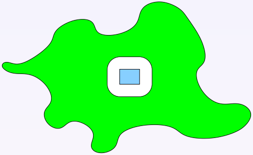
Result: $u|_\omega$ can be approximated in a low-dimensional space.

Generalized interior regularity

Generalization by Hackbusch/Bebendorf (2003), B. (2010):

Let $\tau, \sigma \subseteq \Omega$ with $\operatorname{diam}(\tau) \lesssim \operatorname{dist}(\tau, \sigma)$.
Let $q \in (0, 1)$.
Given $m \in \mathbb{N}$, we find a space $\mathcal{V}_m$ such that

- $\dim(\mathcal{V}_m) \lesssim m^{d+1}$ and
- for each $f \in L^2(\Omega)$ with $\operatorname{supp} f \subseteq \sigma$, we find $\tilde{u} \in \mathcal{V}_m$ such that $\|u - \tilde{u}\|_{L^2(\tau)} \lesssim q^m \|f\|_{L^2(\Omega)}$.

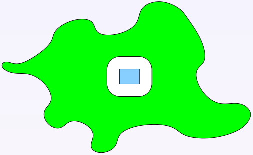


Generalized interior regularity

Generalization by Hackbusch/Bebendorf (2003), B. (2010):

Let $\tau, \sigma \subseteq \Omega$ with $\operatorname{diam}(\tau) \lesssim \operatorname{dist}(\tau, \sigma)$.
Let $q \in (0, 1)$.
Given $m \in \mathbb{N}$, we find a space $\mathcal{V}_m$ such that

- $\dim(\mathcal{V}_m) \lesssim m^{d+1}$ and
- for each $f \in L^2(\Omega)$ with $\operatorname{supp} f \subseteq \sigma$, we find $\tilde{u} \in \mathcal{V}_m$ such that $\|u - \tilde{u}\|_{L^2(\tau)} \lesssim q^m \|f\|_{L^2(\Omega)}$.



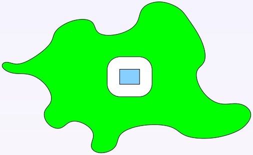
Surprise: Holds even for discontinuous coefficients E.

Generalized interior regularity

Generalization by Hackbusch/Bebendorf (2003), B. (2010):

Let $\tau, \sigma \subseteq \Omega$ with $\operatorname{diam}(\tau) \lesssim \operatorname{dist}(\tau, \sigma)$.
Let $q \in (0, 1)$.
Given $m \in \mathbb{N}$, we find a space $\mathcal{V}_m$ such that

- $\dim(\mathcal{V}_m) \lesssim m^{d+1}$ and
- for each $f \in L^2(\Omega)$ with $\operatorname{supp} f \subseteq \sigma$, we find $\tilde{u} \in \mathcal{V}_m$ such that $\|u - \tilde{u}\|_{L^2(\tau)} \lesssim q^m \|f\|_{L^2(\Omega)}$.



Surprise: Holds even for discontinuous coefficients E.

Goal: Use this property to construct robust preconditioners.

Algebraic version

Linear system $Ax = b$ replaces differential equation $Lu = f$.

Index set $\mathcal{I}$ of localized basis functions replaces domain Ω.

Existence theorem: Let $q \in (0, 1)$. Let $t \subseteq \mathcal{I}$ and let $s \subseteq \mathcal{I}$ satisfy $\operatorname{diam}(t) \lesssim \operatorname{dist}(t, s)$ (in a suitable sense).
Given $m \in \mathbb{N}$, we can find a space $\mathcal{V}_m \subseteq \mathbb{R}^t$ such that

- $\dim(\mathcal{V}_m) \lesssim m^{d+1}$ and
- for each $b \in \mathbb{R}^{\mathcal{I}}$ with $\operatorname{supp} b \subseteq s$, we can find $\tilde{x} \in \mathcal{V}_m$ such that $\|(A^{-1}b)|_t - \tilde{x}\| \lesssim q^m \|b\|$.

Algebraic version

Linear system $Ax = b$ replaces differential equation $Lu = f$.

Index set $\mathcal{I}$ of localized basis functions replaces domain Ω.

Existence theorem: Let $q \in (0, 1)$. Let $t \subseteq \mathcal{I}$ and let $s \subseteq \mathcal{I}$ satisfy diam$(t) \lesssim$ dist(t, s) (in a suitable sense).
Given $m \in \mathbb{N}$, we can find an orthogonal matrix $V_t \in \mathbb{R}^{t\times k}$ such that

- $k \lesssim m^{d+1}$ and
- we have $\|A^{-1}|_{t\times s} - V_t V_t^* A^{-1}|_{t\times s}\| \lesssim q^m$,
  i.e., $A^{-1}|_{t\times s}$ can be approximated by a low-rank matrix.

Algebraic version

Linear system $Ax = b$ replaces differential equation $Lu = f$.

Index set $\mathcal{I}$ of localized basis functions replaces domain Ω.

Existence theorem: Let $q \in (0,1)$. Let $t \subseteq \mathcal{I}$ and let $s \subseteq \mathcal{I}$ satisfy $\operatorname{diam}(t) \lesssim \operatorname{dist}(t,s)$ (in a suitable sense).
Given $m \in \mathbb{N}$, we can find an orthogonal matrix $V_t \in \mathbb{R}^{t \times k}$ such that

- $k \lesssim m^{d+1}$ and
- we have $\|A^{-1}|_{t\times s} - V_t V_t^* A^{-1}|_{t\times s}\| \lesssim q^m$,
  i.e., $A^{-1}|_{t\times s}$ can be approximated by a low-rank matrix.

Questions:

- How to split A^{-1} into submatrices $A^{-1}|_{t\times s}$ we can approximate?
- How to find V_t efficiently?

Overview

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \leq \mathrm{dist}(t, j)\}.$$

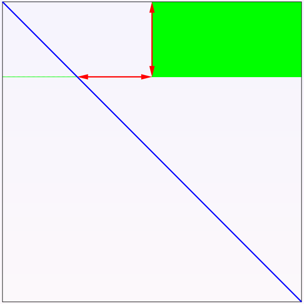
We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \operatorname{diam}(t) \le \operatorname{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



"Row strips" $A^{-1}|_{t \times t^+}$ can be approximated by low rank k:

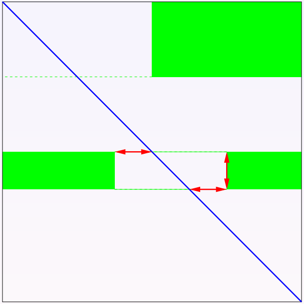
$$A^{-1}|_{t \times t^+} \approx V_t V_t^* A^{-1}|_{t \times t^+}$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \leq \mathrm{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



"Row strips" $A^{-1}|_{t \times t^+}$ can be approximated by low rank k:

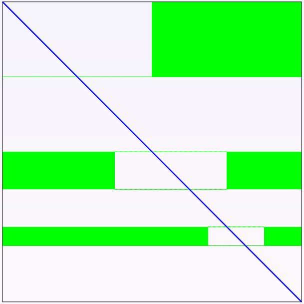
$$A^{-1}|_{t \times t^+} \approx V_t V_t^* A^{-1}|_{t \times t^+}$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \le \mathrm{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



"Row strips" $A^{-1}|_{t \times t^+}$ can be approximated by low rank k:

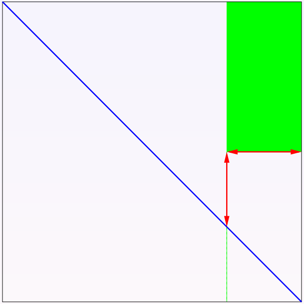
$$A^{-1}|_{t \times t^+} \approx V_t V_t^* A^{-1}|_{t \times t^+}$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \leq \mathrm{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



Due to symmetry, "column strips" $A^{-1}|_{s^+ \times s}$ can also be approximated:

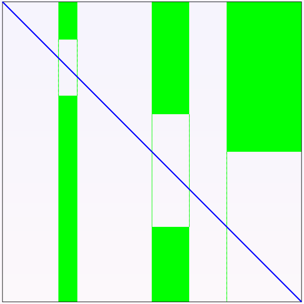
$$A^{-1}|_{s^+ \times s} \approx A^{-1}|_{s^+ \times s} W_s W_s^*$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \leq \mathrm{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



Due to symmetry, "column strips" $A^{-1}|_{s^+ \times s}$ can also be approximated:

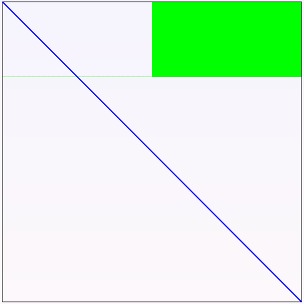
$$A^{-1}|_{s^+ \times s} \approx A^{-1}|_{s^+ \times s} W_s W_s^*$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \leq \mathrm{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



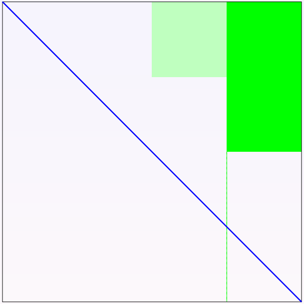
$$A^{-1}|_{t \times t^+} \approx V_t V_t^* A^{-1}|_{t \times t^+}$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \operatorname{diam}(t) \leq \operatorname{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



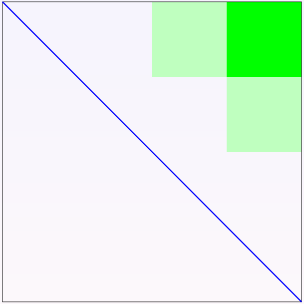
$$A^{-1}|_{t\times t^+} \approx V_t V_t^* A^{-1}|_{t\times t^+}$$
$$A^{-1}|_{s^+\times s} \approx A^{-1}|_{s^+\times s} W_s W_s^*$$

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{ j \in \mathcal{I} \ : \ \mathrm{diam}(t) \leq \mathrm{dist}(t, j) \}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



$$A^{-1}|_{t \times t^+} \approx V_t V_t^* A^{-1}|_{t \times t^+}$$
$$A^{-1}|_{s^+ \times s} \approx A^{-1}|_{s^+ \times s} W_s W_s^*$$
$$A^{-1}|_{t \times s} \approx V_t V_t^* A^{-1}|_{t \times s} W_s W_s^*$$

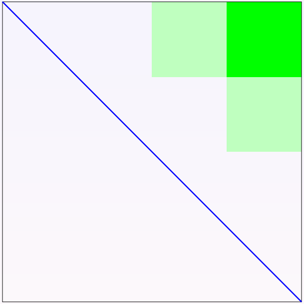
if $t \subseteq s^+$ and $s \subseteq t^+$.

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \;:\; \operatorname{diam}(t) \leq \operatorname{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.



$$\begin{aligned} A^{-1}|_{t\times t^+} &\approx V_t V_t^* A^{-1}|_{t\times t^+} \\ A^{-1}|_{s^+\times s} &\approx A^{-1}|_{s^+\times s} W_s W_s^* \\ A^{-1}|_{t\times s} &\approx V_t V_t^* A^{-1}|_{t\times s} W_s W_s^* \\ &= V_t S_{ts} W_s^* \end{aligned}$$

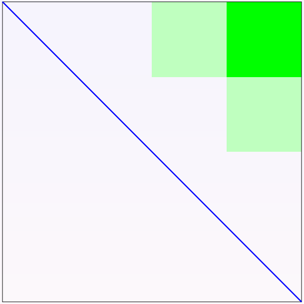
if $t \subseteq s^+$ and $s \subseteq t^+$.

Block representation

Farfield: Given $t \subseteq \mathcal{I}$, denote the farfield of t by

$$t^+ := \{j \in \mathcal{I} \ : \ \mathrm{diam}(t) \le \mathrm{dist}(t, j)\}.$$

We already know that $A^{-1}|_{t \times t^+}$ can be approximated by low rank.

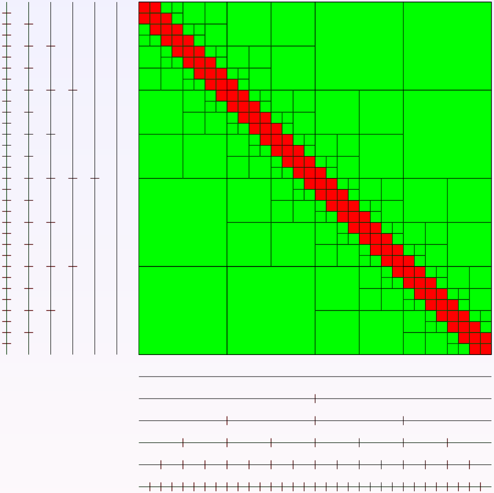


$$\begin{aligned} A^{-1}|_{t\times t^+} &\approx V_t V_t^* A^{-1}|_{t\times t^+} \\ A^{-1}|_{s^+\times s} &\approx A^{-1}|_{s^+\times s} W_s W_s^* \\ A^{-1}|_{t\times s} &\approx V_t V_t^* A^{-1}|_{t\times s} W_s W_s^* \\ &= V_t S_{ts} W_s^* \end{aligned}$$

if $t \subseteq s^+$ and $s \subseteq t^+$.

Result: If $t \times s$ admissible, i.e., if $\max\{\mathrm{diam}(t), \mathrm{diam}(s)\} \le \mathrm{dist}(t, s)$, we can approximate the block $A^{-1}|_{t\times s}$ by a rank-k matrix.

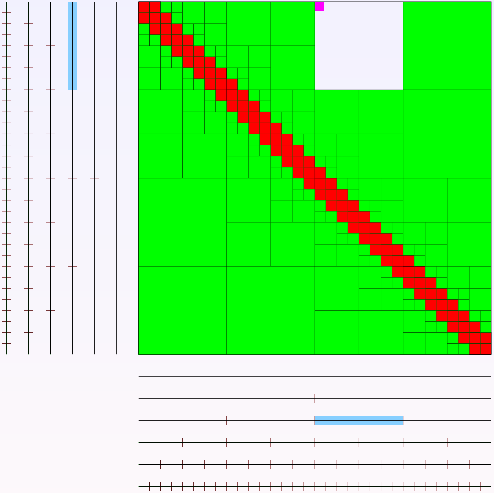
Uniform hierarchical matrix



Split matrix into admissible submatrices $A^{-1}|_{t\times s}$ and a small remainder.

Clusters t, s are organized in a cluster tree.

Uniform hierarchical matrix

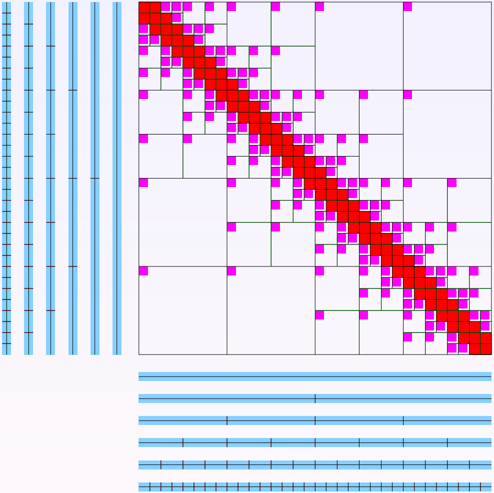


Approximate admissible submatrices.

$$A^{-1}|_{t\times s} \approx V_t S_{ts} W_s^*.$$

Advantage: Coupling matrix S_{ts} is only $k \times k$.

Uniform hierarchical matrix



Approximate admissible submatrices.

$$A^{-1}|_{t\times s} \approx V_t S_{ts} W_s^*.$$

Advantage: Coupling matrix S_{ts} is only $k \times k$.

Storage required:
$\mathcal{O}(nk)$ for coupling matrices,
$\mathcal{O}(nk \log n)$ for cluster bases (V_t) and (W_s).

Cluster basis

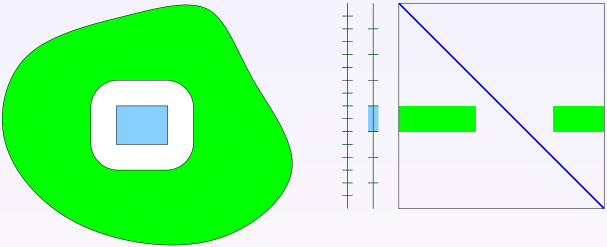
Problem: Storage requirements $\sim nk \log n$ for (V_t) too high.

Idea: Establish hierarchical structure of the cluster basis (V_t).

Cluster basis

Problem: Storage requirements $\sim nk \log n$ for (V_t) too high.

Idea: Establish hierarchical structure of the cluster basis (V_t).



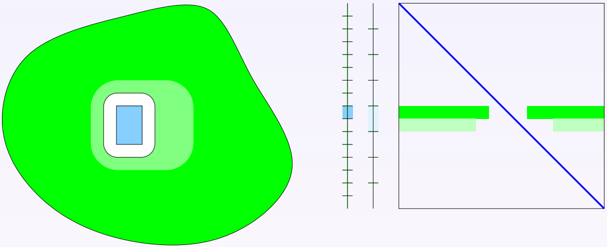
Hierarchical relationship between a cluster t

$$A^{-1}|_{t\times t^+} \approx V_t V_t^* A^{-1}|_{t\times t^+},$$

Cluster basis

Problem: Storage requirements $\sim nk \log n$ for (V_t) too high.

Idea: Establish hierarchical structure of the cluster basis (V_t).



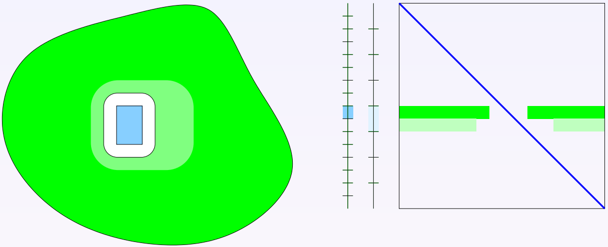
Hierarchical relationship between a cluster t and its son $r \subseteq t$:

$$A^{-1}|_{t\times t^+} \approx V_t V_t^* A^{-1}|_{t\times t^+}, \qquad A^{-1}|_{r\times r^+} \approx V_r V_r^* A^{-1}|_{r\times r^+}.$$

Cluster basis

Problem: Storage requirements $\sim nk \log n$ for (V_t) too high.

Idea: Establish hierarchical structure of the cluster basis (V_t).



Hierarchical relationship between a cluster t and its son $r \subseteq t$:

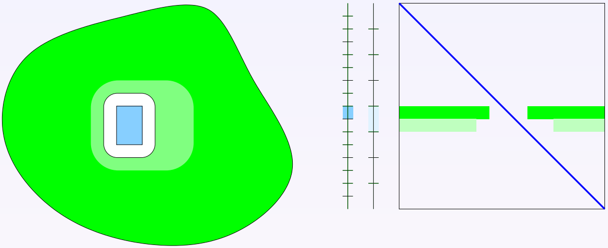
$$A^{-1}|_{t\times t^+} \approx V_t V_t^* A^{-1}|_{t\times t^+}, \qquad A^{-1}|_{r\times r^+} \approx V_r V_r^* A^{-1}|_{r\times r^+}.$$

V_r should be able to approximate the "upper half" of V_t.

Cluster basis

Problem: Storage requirements $\sim nk \log n$ for (V_t) too high.

Idea: Establish hierarchical structure of the cluster basis (V_t).

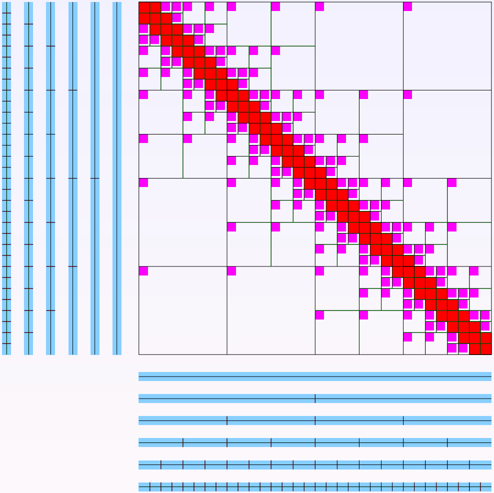


Hierarchical relationship between a cluster t and its son $r \subseteq t$:

$$V_t|_{r \times k} = V_r E_r$$

holds with a $k \times k$ transfer matrix E_r.

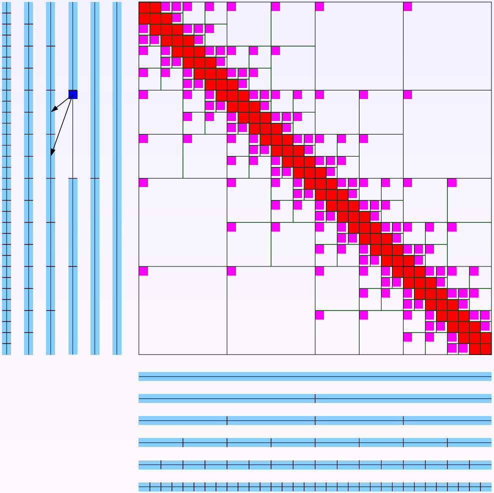
$\mathcal{H}^2$-matrix



Blocks approximated by

$$A^{-1}|_{t\times s} \approx V_t S_{ts} W_s^*$$

$\mathcal{H}^2$-matrix



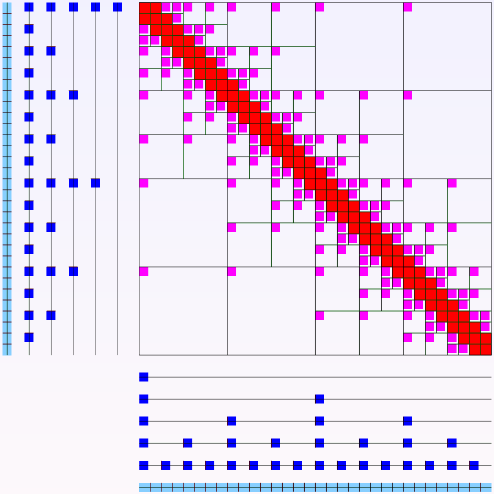
Blocks approximated by

$$A^{-1}|_{t\times s} \approx V_t S_{ts} W_s^*$$

Cluster basis nested:

$$V_t = \begin{pmatrix} V_{t_1} E_{t_1} \\ V_{t_2} E_{t_2} \end{pmatrix}$$

$\mathcal{H}^2$-matrix



Blocks approximated by

$$A^{-1}|_{t\times s} \approx V_t S_{ts} W_s^*$$

Cluster basis nested:

$$V_t = \begin{pmatrix} V_{t_1} E_{t_1} \\ V_{t_2} E_{t_2} \end{pmatrix}$$

Result: Complexity $\mathcal{O}(nk)$

Prior work

Integral operators: Discretization of integral operators with asymptotically smooth kernel function leads immediately to $\mathcal{H}^2$-matrices.

Compression: Any $n \times n$ matrix G can be approximated by a quasi-optimal $\mathcal{H}^2$-matrix in $\mathcal{O}(n^2 k)$ operations.
Faster algorithm for $\mathcal{H}$-matrices takes $\mathcal{O}(nk^2 \log n)$ operations.

Algebraic operations: For fixed cluster bases, best approximation of the product of two $\mathcal{H}^2$-matrices can be computed in $\mathcal{O}(nk^2)$ operations.
Inversion algorithm also explored, reaches only low accuracy.

Overview

Preconditioners

Remember: Inverse of A can be approximated even if coefficients of the PDE are discontinuous or anisotropic.
$\rightarrow$ Robust approach for large family of problems.

Inversion approach: Approximate the inverse $X := A^{-1}$ by $\widetilde{X}$ such that $\|I - \widetilde{X}A\| \leq \varrho$ for a moderate $\varrho < 1$.

Factorization approach: Approximate LR factorization $\widetilde{L}\widetilde{R} \approx A$, evaluate $\widetilde{X} := \widetilde{R}^{-1}\widetilde{L}^{-1}$ by forward and backward substitution.

Iteration: Simple Richardson iteration

$$x^{(m+1)} \leftarrow x^{(m)} - \widetilde{X}(Ax^{(m)} - b)$$

converges at rate $\leq \varrho$, Krylov methods usually faster.

Inversion

Goal: Given an $\mathcal{H}^2$-matrix A, compute its inverse A^{-1}.

Approach: If A is inadmissible, compute A^{-1} directly.
Otherwise, A consists of submatrices

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and computing a partial block LR factorization yields

Inversion

Goal: Given an $\mathcal{H}^2$-matrix A, compute its inverse A^{-1}.

Approach: If A is inadmissible, compute A^{-1} directly.
Otherwise, A consists of submatrices

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and computing a partial block LR factorization yields

$$\begin{pmatrix} I & \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix}.$$

Inversion

Goal: Given an $\mathcal{H}^2$-matrix A, compute its inverse A^{-1}.

Approach: If A is inadmissible, compute A^{-1} directly.
Otherwise, A consists of submatrices

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and computing a partial block LR factorization yields

$$\begin{pmatrix} I & \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ & S \end{pmatrix}.$$

with the Schur complement $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$.

Inversion

Goal: Given an $\mathcal{H}^2$-matrix A, compute its inverse A^{-1}.

Approach: If A is inadmissible, compute A^{-1} directly.
Otherwise, A consists of submatrices

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and computing a partial block LR factorization yields

$$\begin{pmatrix} A_{11}^{-1} & -A_{11}^{-1} A_{12} S^{-1} \\ & S^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -A_{21} A_{11}^{-1} & I \end{pmatrix} = A^{-1}$$

with the Schur complement $S = A_{22} - A_{21} A_{11}^{-1} A_{12}$.

Inversion

Goal: Given an $\mathcal{H}^2$-matrix A, compute its inverse A^{-1}.

Approach: If A is inadmissible, compute A^{-1} directly.
Otherwise, A consists of submatrices

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and computing a partial block LR factorization yields

$$\begin{pmatrix} A_{11}^{-1} & -A_{11}^{-1} A_{12} S^{-1} \\ & S^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -A_{21} A_{11}^{-1} & I \end{pmatrix} = A^{-1}$$

with the Schur complement $S = A_{22} - A_{21} A_{11}^{-1} A_{12}$.

Result: If we have an efficient algorithm for computing $Z \leftarrow Z + \alpha XY$, we can compute the inverse by recursion.

LR factorization

Goal: Given an $\mathcal{H}^2$-matrix A, compute its LR factorization $A = LR$.

Approach: If A is inadmissible, compute $A = LR$ directly.
Otherwise, use submatrices

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$$

LR factorization

Goal: Given an $\mathcal{H}^2$-matrix A, compute its LR factorization $A = LR$.

Approach: If A is inadmissible, compute $A = LR$ directly.
Otherwise, use submatrices

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$$

This is equivalent to

$$A_{11} = L_{11} R_{11},$$
$$A_{12} = L_{11} R_{12}, \qquad A_{21} = L_{21} R_{11},$$
$$A_{22} - L_{21} R_{12} = L_{22} R_{22}.$$

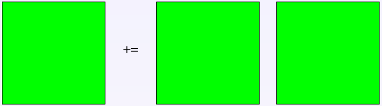
If we can compute $Z \leftarrow Z + \alpha XY$ efficiently, we can use recursion to perform matrix forward substitution and find the LR factorization.

Overview

Multiplication

Goal: Perform update $Z|_{t \times r} \leftarrow Z|_{t \times r} + \alpha X|_{t \times s} Y|_{s \times r}$ efficiently.

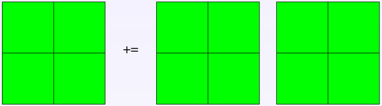
Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

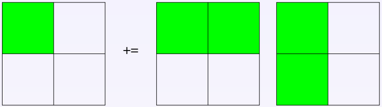
Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

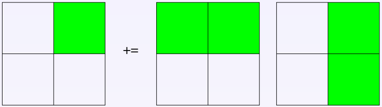
Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

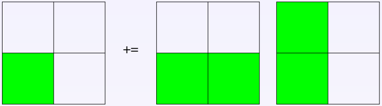
Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

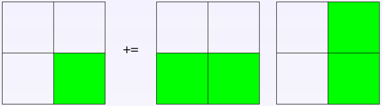
Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

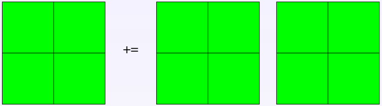
Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

Recursion: If both (t, s) and (s, r) are not admissible, handle submatrices by recursion.



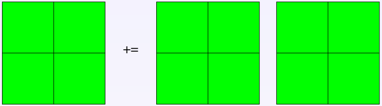
Idea: If (s, r) is admissible, we have

$$X|_{t\times s} Y|_{s\times r} = \; X|_{t\times s} V_s \; S_{s,r} W_r^*$$

Multiplication

Goal: Perform update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha X|_{t\times s} Y|_{s\times r}$ efficiently.

Recursion: If both (t,s) and (s,r) are not admissible, handle submatrices by recursion.



Idea: If (s,r) is admissible, we have

$$X|_{t\times s} Y|_{s\times r} = (X|_{t\times s} V_s) S_{s,r} W_r^*$$

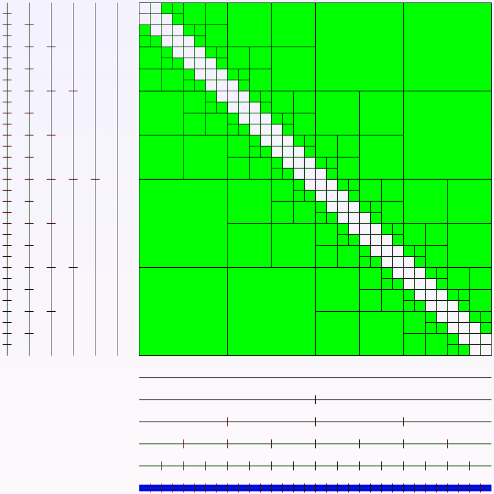
and can split the computation into two steps:

- Column-wise matrix-vector multiplications yield $A_{t,s} = X|_{t\times s} V_s$
- Local low-rank update $Z|_{t\times r} \leftarrow Z|_{t\times r} + \alpha A_{t,s} S_{s,r} W_r^*$

Similar approach can be used if (t,s) is admissible.

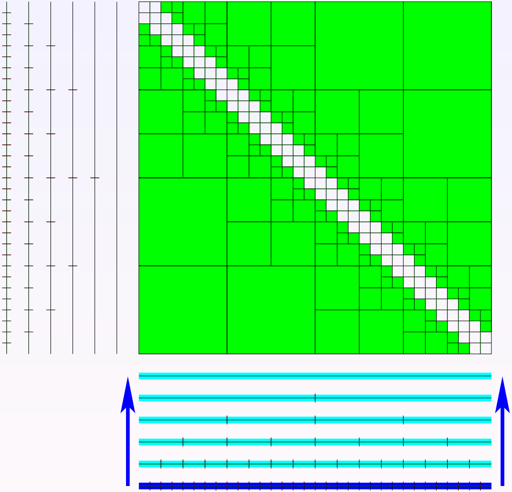
Matrix-vector multiplication

Goal: Compute $z \leftarrow z + \alpha X y$, $X|_{t\times s} = V_t S_b W_s^*$ for admissible $t \times s$



Matrix-vector multiplication

Goal: Compute $z \leftarrow z + \alpha X y$, $X|_{t \times s} = V_t S_b W_s^*$ for admissible $t \times s$



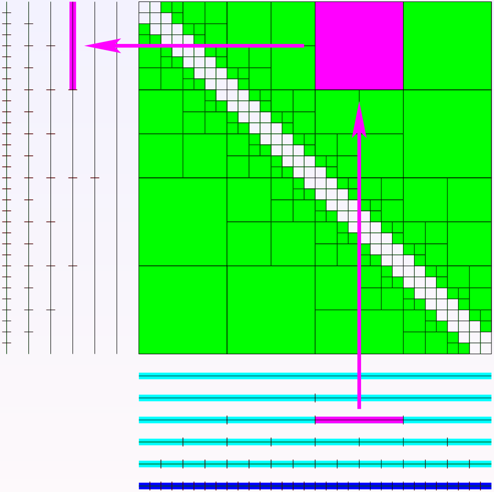
Forward step: For all s, compute $\hat{y}_s = W_s^* y|_s$

Non-leaf clusters: Use transfer matrices

$$\hat{y}_s = \begin{pmatrix} W_{s_1} F_{s_1} \\ W_{s_2} F_{s_2} \end{pmatrix}^* \begin{pmatrix} y|_{\hat{s}_1} \\ y|_{\hat{s}_2} \end{pmatrix} = F_{s_1}^* \hat{y}_{s_1} + F_{s_2}^* \hat{y}_{s_2}$$

Matrix-vector multiplication

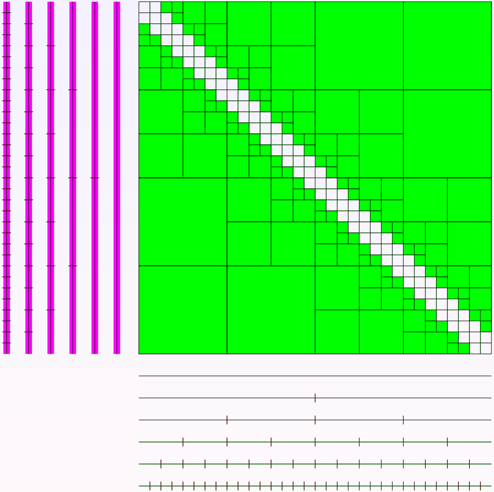
Goal: Compute $z \leftarrow z + \alpha X y$, $X|_{t \times s} = V_t S_b W_s^*$ for admissible $t \times s$



Interaction step: For all blocks $t \times s$, compute $\hat{z}_t \leftarrow \hat{z}_t + S_{t,s} \hat{y}_s$

Matrix-vector multiplication

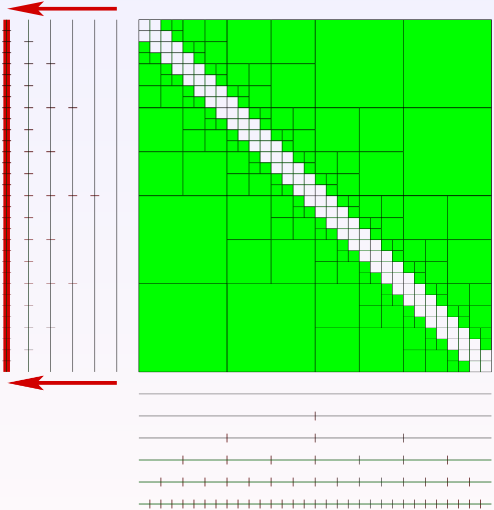
Goal: Compute $z \leftarrow z + \alpha X y$, $X|_{t \times s} = V_t S_b W_s^*$ for admissible $t \times s$



Interaction step: For all blocks $t \times s$, compute $\hat{z}_t \leftarrow \hat{z}_t + S_{t,s} \hat{y}_s$

Matrix-vector multiplication

Goal: Compute $z \leftarrow z + \alpha X y$, $X|_{t \times s} = V_t S_b W_s^*$ for admissible $t \times s$



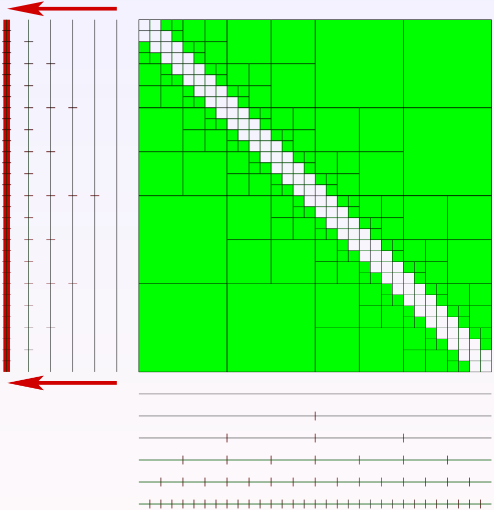
Backward step: For all t, compute $z|_t \leftarrow z|_t + V_t \hat{z}_t$

Non-leaf clusters:
Use transfer matrices to shift coefficients to the leaves

$$\hat{z}_{t_1} \leftarrow \hat{z}_{t_1} + E_{t_1} \hat{z}_t,$$
$$\hat{z}_{t_2} \leftarrow \hat{z}_{t_2} + E_{t_2} \hat{z}_t.$$

Matrix-vector multiplication

Goal: Compute $z \leftarrow z + \alpha X y$, $X|_{t\times s} = V_t S_b W_s^*$ for admissible $t \times s$

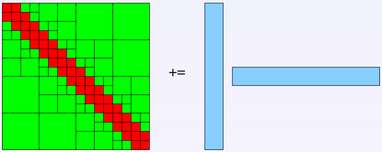


Result:
$\mathcal{O}(k(\#t + \#s))$ operations for the entire computation.

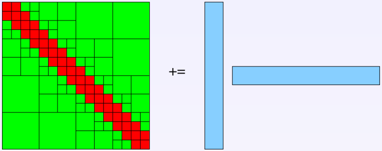
$A_{t,s} = X|_{t\times s} V_s$ computed in $\mathcal{O}(k^2(\#t + \#s))$ operations.

Low-rank update

Goal: Approximate $Z \leftarrow Z + AB^*$.



Low-rank update



Goal: Approximate $Z \leftarrow Z + AB^*$.

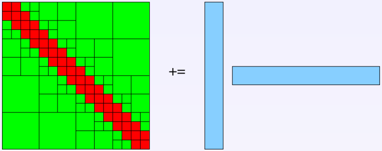
Idea: Construct a new $\mathcal{H}^2$-matrix $\widehat{Z} = Z + AB^*$ using

$$\widehat{V}_t = \begin{pmatrix} V_t & A|_{t\times k} \end{pmatrix}, \qquad \widehat{W}_s = \begin{pmatrix} W_s & B|_{s\times k} \end{pmatrix}, \qquad \widehat{S}_b = \begin{pmatrix} S_b & \\ & I \end{pmatrix},$$

$$\widehat{Z}|_{t\times s} = \widehat{V}_t \widehat{S}_b \widehat{W}_s^* = \begin{pmatrix} V_t & A|_{t\times k} \end{pmatrix} \begin{pmatrix} S_b & \\ & I \end{pmatrix} \begin{pmatrix} W_s^* \\ B|_{s\times k}^* \end{pmatrix} = (Z + AB^*)|_{t\times s}.$$

Low-rank update

Goal: Approximate $Z \leftarrow Z + AB^*$.



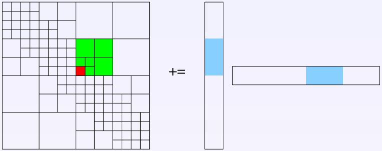
Idea: Construct a new $\mathcal{H}^2$-matrix $\widehat{Z} = Z + AB^*$ using

$$\widehat{V}_t = \begin{pmatrix} V_t & A|_{t\times k} \end{pmatrix}, \qquad \widehat{W}_s = \begin{pmatrix} W_s & B|_{s\times k} \end{pmatrix}, \qquad \widehat{S}_b = \begin{pmatrix} S_b & \\ & I \end{pmatrix},$$

$$\widehat{Z}|_{t\times s} = \widehat{V}_t \widehat{S}_b \widehat{W}_s^* = \begin{pmatrix} V_t & A|_{t\times k} \end{pmatrix} \begin{pmatrix} S_b & \\ & I \end{pmatrix} \begin{pmatrix} W_s^* \\ B|_{s\times k}^* \end{pmatrix} = (Z + AB^*)|_{t\times s}.$$

Algorithm: Use singular value decompositions to reduce the rank of the exact result $\widehat{Z}$ and obtain the approximation $\widetilde{Z} \approx Z + AB^*$.
$\rightarrow$ Complexity $\mathcal{O}(k^2 n)$, error can be controlled.

Low-rank update



Goal: Approximate $Z \leftarrow Z + AB^*$.

Idea: Construct a new $\mathcal{H}^2$-matrix $\widehat{Z} = Z + AB^*$ using

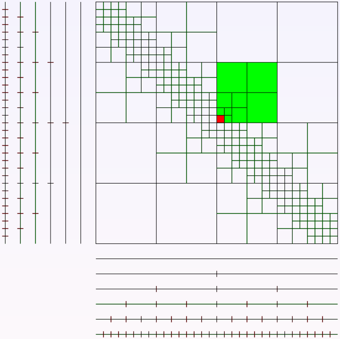
$$\widehat{V}_t = \begin{pmatrix} V_t & A|_{t\times k} \end{pmatrix}, \qquad \widehat{W}_s = \begin{pmatrix} W_s & B|_{s\times k} \end{pmatrix}, \qquad \widehat{S}_b = \begin{pmatrix} S_b & \\ & I \end{pmatrix},$$

$$\widehat{Z}|_{t\times s} = \widehat{V}_t \widehat{S}_b \widehat{W}_s^* = \begin{pmatrix} V_t & A|_{t\times k} \end{pmatrix} \begin{pmatrix} S_b & \\ & I \end{pmatrix} \begin{pmatrix} W_s^* \\ B|_{s\times k}^* \end{pmatrix} = (Z + AB^*)|_{t\times s}.$$

Algorithm: Use singular value decompositions to reduce the rank of the exact result $\widehat{Z}$ and obtain the approximation $\widetilde{Z} \approx Z + AB^*$.
$\rightarrow$ Complexity $\mathcal{O}(k^2 n)$, error can be controlled.

Local low-rank update

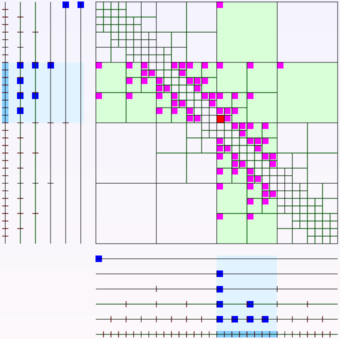
Goal: Approximate $Z|_{t\times r} \leftarrow Z|_{t\times r} + AB^*$ with $A \in \mathbb{R}^{t\times k}$, $B \in \mathbb{R}^{s\times k}$.



Local low-rank update

Goal: Approximate $Z|_{t \times r} \leftarrow Z|_{t \times r} + AB^*$ with $A \in \mathbb{R}^{t \times k}$, $B \in \mathbb{R}^{s \times k}$.

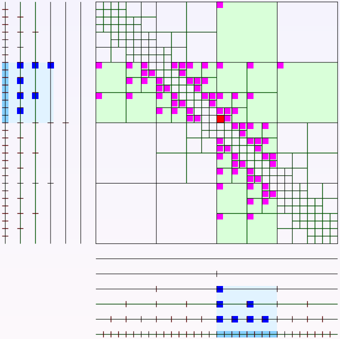
Problem: If we have to change the cluster bases, all blocks intersecting t or s are affected.



Local low-rank update

Goal: Approximate $Z|_{t\times r} \leftarrow Z|_{t\times r} + AB^*$ with $A \in \mathbb{R}^{t\times k}$, $B \in \mathbb{R}^{s\times k}$.

Problem: If we have to change the cluster bases, all blocks intersecting t or s are affected.

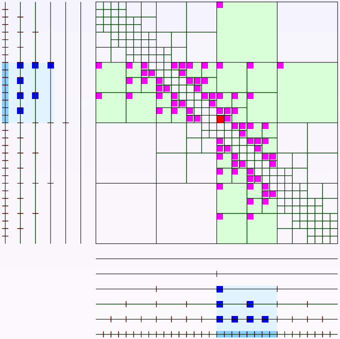


Observation: Due to the nested structure, changing one transfer matrix changes the basis for all ancestors.
$\rightarrow$ Only work on descendants of t and s.

Local low-rank update

Goal: Approximate $Z|_{t\times r} \leftarrow Z|_{t\times r} + AB^*$ with $A \in \mathbb{R}^{t\times k}$, $B \in \mathbb{R}^{s\times k}$.

Problem: If we have to change the cluster bases, all blocks intersecting t or s are affected.



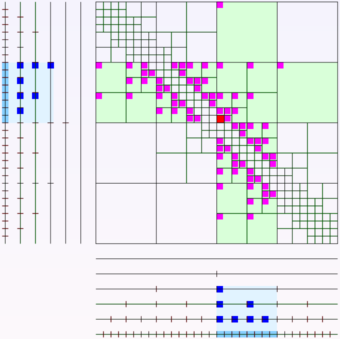
Observation: Due to the nested structure, changing one transfer matrix changes the basis for all ancestors.
$\rightarrow$ Only work on descendants of t and s.

Important: Since ancestors are effected by new cluster bases, they have to be taken into account.
$\rightarrow$ Handled by weight matrices.

Local low-rank update

Goal: Approximate $Z|_{t\times r} \leftarrow Z|_{t\times r} + AB^*$ with $A \in \mathbb{R}^{t\times k}$, $B \in \mathbb{R}^{s\times k}$.

Problem: If we have to change the cluster bases, all blocks intersecting t or s are affected.



Observation: Due to the nested structure, changing one transfer matrix changes the basis for all ancestors.
$\rightarrow$ Only work on descendants of t and s.

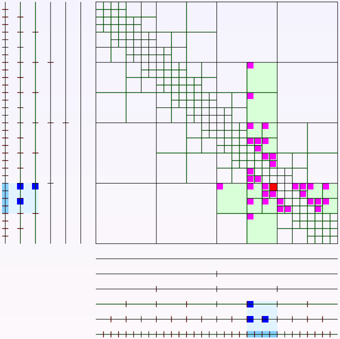
Important: Since ancestors are effected by new cluster bases, they have to be taken into account.
$\rightarrow$ Handled by weight matrices.

Result: Complexity $\mathcal{O}(k^2(\#t + \#s))$

Local low-rank update

Goal: Approximate $Z|_{t\times r} \leftarrow Z|_{t\times r} + AB^*$ with $A \in \mathbb{R}^{t\times k}$, $B \in \mathbb{R}^{s\times k}$.

Problem: If we have to change the cluster bases, all blocks intersecting t or s are affected.



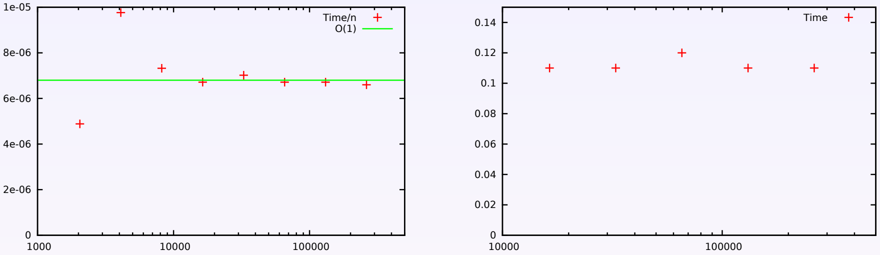
Observation: Due to the nested structure, changing one transfer matrix changes the basis for all ancestors.
$\rightarrow$ Only work on descendants of t and s.

Important: Since ancestors are effected by new cluster bases, they have to be taken into account.
$\rightarrow$ Handled by weight matrices.

Result: Complexity $\mathcal{O}(k^2(\#t + \#s))$

Experiment: Local low-rank update

Goal: Low-rank update to entire matrix (left) or submatrix (right).

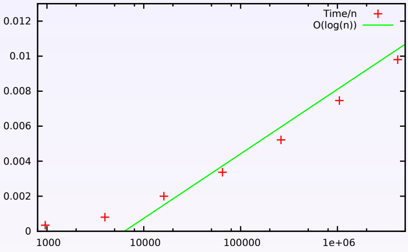


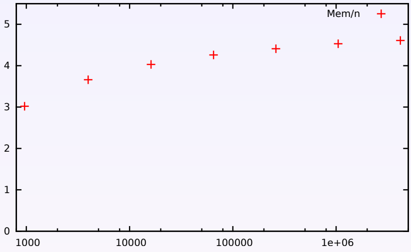
Results:

- Updating the entire matrix takes $\sim nk^2$ operations.
- Updating a submatrix $Z|_{t\times s}$ takes $\sim (\#t + \#s)k^2$ operations, independent of total matrix size n.

Experiment: FEM LR decomposition

Goal: Approximate the LR decomposition of a FEM stiffness matrix.



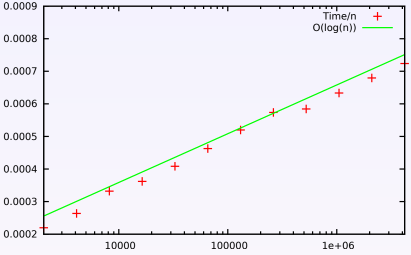


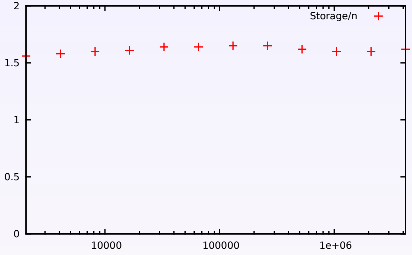
Results:

- Accuracy $\|I - \widetilde{R}^{-1}\widetilde{L}^{-1}A\|_2 \approx 0.01$.
- Factorization in $\sim n \log n$ operations.
- Storage requirements $\sim n$.

Experiment: BEM LR decomposition

Goal: Approximate the LR decomposition of a BEM stiffness matrix.





Results:

- Accuracy $\|I - \widetilde{R}^{-1}\widetilde{L}^{-1}A\|_2 \approx 0.02$.
- Factorization in $\sim n \log n$ operations.
- Storage requirements $\sim n$.

Overview

Matrix equations

Observation: Inverse and LR factorization given by matrix equations

$$AX = I, \qquad LR = A.$$

Goal: Handle more general matrix equations, e.g.,

- $AX + XB = C$ (Sylvester's equation),
- $A^* X + XA - XBX = C$ (Riccati's equation),
- $AXA = C$ (certain stochastic PDEs).

Approach: Look for an $\mathcal{H}^2$-matrix approximation $\widetilde{X}$.

Matrix Galerkin

Example: Consider model problem $AX = B$.

Question: How to construct $\mathcal{H}^2$-matrix approximation $\widetilde{X}$ efficiently?

Idea: Variational formulation :

$$\langle Y, AX \rangle_F = \langle Y, B \rangle_F \qquad \text{for all } Y \in \mathbb{R}^{n \times n}.$$

Matrix Galerkin

Example: Consider model problem $AX = B$.

Question: How to construct $\mathcal{H}^2$-matrix approximation $\widetilde{X}$ efficiently?

Idea: Variational formulation and Galerkin's method:

$$\langle \widetilde{Y}, A\widetilde{X} \rangle_F = \langle \widetilde{Y}, B \rangle_F \qquad \text{for all } \widetilde{Y} \in \mathcal{H}_2.$$

Matrix Galerkin

Example: Consider model problem $AX = B$.

Question: How to construct $\mathcal{H}^2$-matrix approximation $\widetilde{X}$ efficiently?

Idea: Variational formulation and Galerkin's method:

$$\langle \widetilde{Y}, A\widetilde{X} \rangle_F = \langle \widetilde{Y}, B \rangle_F \qquad \text{for all } \widetilde{Y} \in \mathcal{H}_2.$$

Important: Spectral properties of A carry over to bilinear form:

$$\begin{aligned} & \alpha \|y\|_2^2 \leq \langle y, Ay \rangle_2 \leq \beta \|y\|_2^2 && \text{for all } y \in \mathbb{R}^{\mathcal{I}} \\ \iff \quad & \alpha \|Y\|_F^2 \leq \langle Y, AY \rangle_F \leq \beta \|Y\|_F^2 && \text{for all } Y \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}. \end{aligned}$$

Result: If A is s.p.d., $\|X - \widetilde{X}\|_F$ can be estimated by Céa's lemma.

Linear system

Stiffness matrix requires a basis of the space $\mathcal{H}^2$.

Simple approach based on admissible blocks:

$$\widetilde{X} = \sum_{t \times s} V_t S_{ts} W_s^*$$

Linear system

Stiffness matrix requires a basis of the space $\mathcal{H}^2$.

Simple approach based on admissible blocks:

$$\widetilde{X} = \sum_{t\times s} V_t S_{ts} W_s^*$$

Result: System of equations

$$\langle V_p S_{pq} W_q^*, A\widetilde{X}\rangle_F = \langle V_p S_{pq} W_q^*, B\rangle_F \quad \text{for all blocks } p \times q,\ S_{pq} \in \mathbb{R}^{k\times k}$$

Linear system

Stiffness matrix requires a basis of the space $\mathcal{H}^2$.

Simple approach based on admissible blocks:

$$\widetilde{X} = \sum_{t \times s} V_t S_{ts} W_s^*$$

Result: System of equations

$$\langle S_{pq}, V_p^* A \widetilde{X} W_q \rangle_F = \langle S_{pq}, V_p^* B W_q \rangle_F \quad \text{for all blocks } p \times q,\ S_{pq} \in \mathbb{R}^{k \times k}$$

Linear system

Stiffness matrix requires a basis of the space $\mathcal{H}^2$.

Simple approach based on admissible blocks:

$$\widetilde{X} = \sum_{t \times s} V_t S_{ts} W_s^*$$

Result: System of equations

$$V_p^* A \widetilde{X} W_q = V_p^* B W_q \qquad \text{for all blocks } p \times q.$$

Linear system

Stiffness matrix requires a basis of the space $\mathcal{H}^2$.

Simple approach based on admissible blocks:

$$\widetilde{X} = \sum_{t \times s} V_t S_{ts} W_s^*$$

Result: System of equations

$$\sum_{t \times s} V_p^* A V_t S_{ts} W_s^* W_q = V_p^* B W_q \qquad \text{for all blocks } p \times q.$$

Goal: Find matrices S_{ts} solving this system.

Linear system

Stiffness matrix requires a basis of the space $\mathcal{H}^2$.

Simple approach based on admissible blocks:

$$\widetilde{X} = \sum_{t \times s} V_t S_{ts} W_s^*$$

Result: System of equations

$$\sum_{t \times s} V_p^* A V_t S_{ts} W_s^* W_q = V_p^* B W_q \qquad \text{for all blocks } p \times q.$$

Goal: Find matrices S_{ts} solving this system.

Challenges: Sparse system? Fast solver?

Sparsity

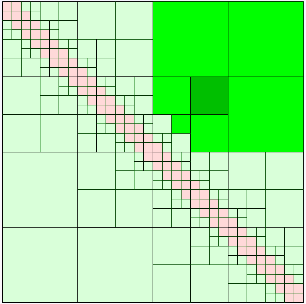
Question: How many blocks $t \times s$ contribute to a "matrix row" $p \times q$?

$$\sum_{t \times s} V_p^* A V_t S_{ts} W_s^* W_q = V_p^* B W_q$$

Sparsity

Question: How many blocks $t \times s$ contribute to a "matrix row" $p \times q$?

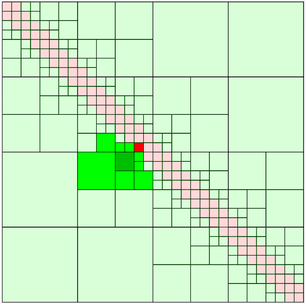
$$\sum_{t \times s} V_p^* A V_t S_{ts} W_s^* W_q = V_p^* B W_q \qquad \rightarrow \qquad V_p^* A V_t \neq 0, \; W_s^* W_q \neq 0$$



Sparsity

Question: How many blocks $t \times s$ contribute to a "matrix row" $p \times q$?

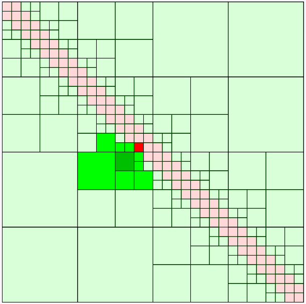
$$\sum_{t \times s} V_p^* A V_t S_{ts} W_s^* W_q = V_p^* B W_q \qquad \rightarrow \qquad V_p^* A V_t \neq 0,\ \ W_s^* W_q \neq 0$$



Sparsity

Question: How many blocks $t \times s$ contribute to a "matrix row" $p \times q$?

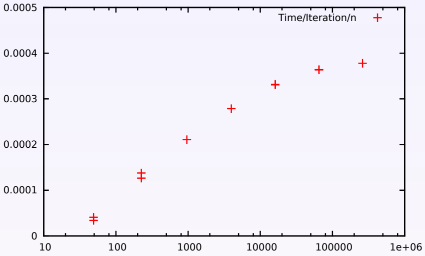
$$\sum_{t \times s} V_p^* A V_t S_{ts} W_s^* W_q = V_p^* B W_q \qquad \rightarrow \qquad V_p^* A V_t \neq 0, \; W_s^* W_q \neq 0$$

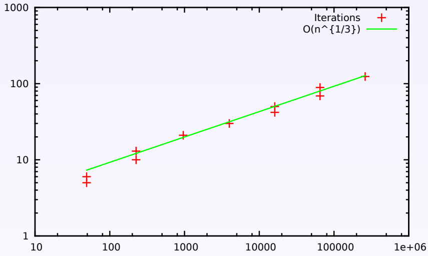


Result: The number of relevant blocks is bounded, and these blocks can be computed efficiently.

Experiment: Matrix Galerkin inverse

Goal: Approximate solution of $AX = I$ by the matrix Galerkin method, where A is a FE stiffness matrix for a two-dimensional domain.





Results:

- Accuracy $\|I - A\widetilde{X}\|_2 \approx 0.1$.
- Iteration steps require $\sim n$ operations.
- cg method apparently requires $\sim n^{1/3}$ iterations.

Conclusion

$\mathcal{H}^2$-matrix preconditioners of accuracy $\|I - BA\|_2 \approx 0.01$:

- Time $\sim n \log n$.
- Storage $\sim n$.

$\mathcal{H}^2$-matrix algebraic operations:

- Multiplication, inversion, LR factorization: $O(nk^2 \log n)$.
- Local low-rank update of $t \times s$ submatrix: $O(k^2(\#t + \#s))$.

$\mathcal{H}^2$-matrix Galerkin:

- Applicable to (moderately) general matrix equations.
- $O(nk^2)$ operations per iteration step.