Hierarchical vectors

Steffen Börm

University of Kiel, Germany

Conference in Honor of Volker Mehrmann
Numerical Algebra, Matrix Theory, DAEs, and Control Theory,
Berlin, 7th of May, 2015

Overview

Overview

Data-sparse representations

Challenge: Non-local operators appear naturally in many applications.

- Solution operators of partial differential equations.
- Boundary integral method.
- Population dynamics.
- Control theory.

Discretization schemes lead to highly non-sparse matrices.

Data-sparse representations

Challenge: Non-local operators appear naturally in many applications.

- Solution operators of partial differential equations.
- Boundary integral method.
- Population dynamics.
- Control theory.

Discretization schemes lead to highly non-sparse matrices.

Idea: Represent these matrices in a data-sparse representation, i.e., by $\mathcal{O}(n \log^\alpha(n))$ coefficients instead of n^2.

Examples: Panel clustering, multipole expansion, wavelets, hierarchical matrices.

Low-rank factorization

Submatrices $G|_{t\times s}$ are approximated by low-rank factorizations

$$G|_{t\times s} \approx Q_t S_{ts} Q_s^*$$

with small coupling matrices $S_{ts} \in \mathbb{R}^{k\times k}$.

Cluster basis matrices $Q_t \in \mathbb{R}^{t\times k}$ represented in nested hierarchy

$$Q_t|_{t'\times k} = Q_{t'} E_{t'}$$

with small transfer matrices $E_{t'} \in \mathbb{R}^{k\times k}$.

Standard example: Polynomial interpolation,

$$g(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \ell_{t,\nu}(x) g(\xi_{t,\nu}, \xi_{s,\mu}) \ell_{s,\mu}(y), \quad \ell_{t,\nu} = \sum_{\mu=1}^{k} \ell_{t,\nu}(\xi_{t',\mu}) \ell_{t',\mu}.$$

Low-rank factorization

Submatrices $G|_{t\times s}$ are approximated by low-rank factorizations

$$G|_{t\times s} \approx Q_t S_{ts} Q_s^*$$

with small coupling matrices $S_{ts} \in \mathbb{R}^{k\times k}$.

Cluster basis matrices $Q_t \in \mathbb{R}^{t\times k}$ represented in nested hierarchy

$$Q_t|_{t'\times k} = Q_{t'} E_{t'}$$

with small transfer matrices $E_{t'} \in \mathbb{R}^{k\times k}$.

Standard example: Polynomial interpolation,

$$g(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \ell_{t,\nu}(x) g(\xi_{t,\nu}, \xi_{s,\mu}) \ell_{s,\mu}(y), \quad \ell_{t,\nu} = \sum_{\mu=1}^{k} \ell_{t,\nu}(\xi_{t',\mu}) \ell_{t',\mu}.$$

Low-rank factorization

Submatrices $G|_{t\times s}$ are approximated by low-rank factorizations

$$G|_{t\times s} \approx Q_t S_{ts} Q_s^*$$

with small coupling matrices $S_{ts} \in \mathbb{R}^{k\times k}$.

Cluster basis matrices $Q_t \in \mathbb{R}^{t\times k}$ represented in nested hierarchy

$$Q_t|_{t'\times k} = Q_{t'} E_{t'}$$

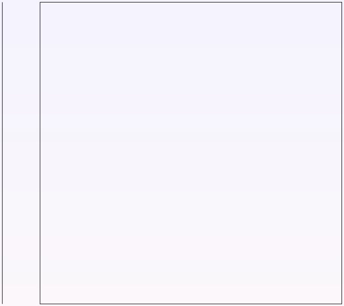
with small transfer matrices $E_{t'} \in \mathbb{R}^{k\times k}$.

Standard example: Polynomial interpolation,

$$g(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} \ell_{t,\nu}(x) g(\xi_{t,\nu}, \xi_{s,\mu}) \ell_{s,\mu}(y), \quad \ell_{t,\nu} = \sum_{\mu=1}^{k} \ell_{t,\nu}(\xi_{t',\mu}) \ell_{t',\mu}.$$

$\mathcal{H}^2$-matrix

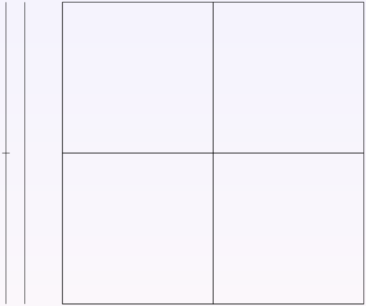
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

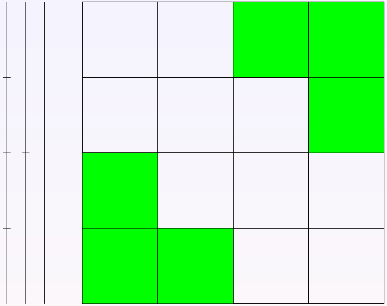
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

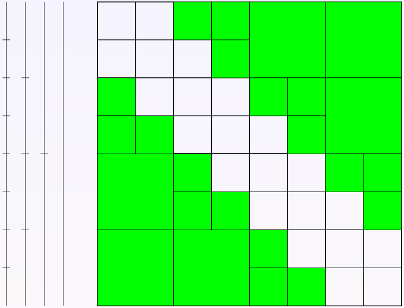
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

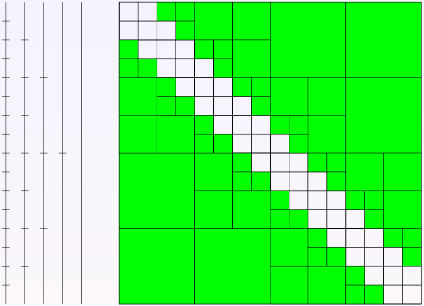
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

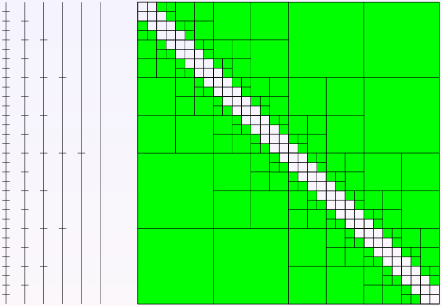
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

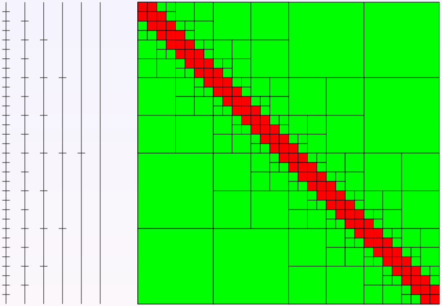
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

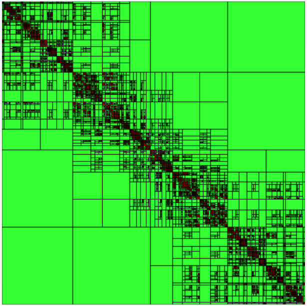
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

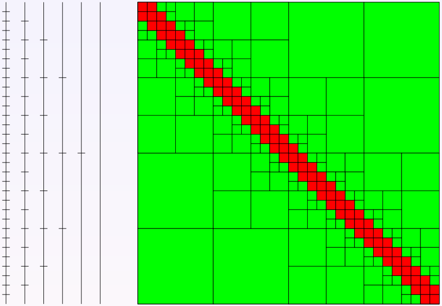
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

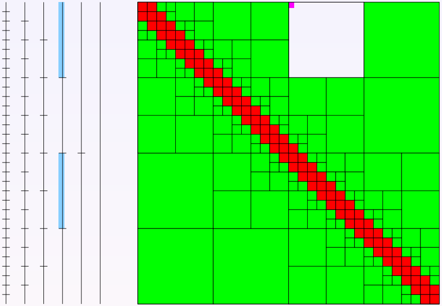
Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

$\mathcal{H}^2$-matrix

Idea: Split matrix into submatrices that have numerically low rank.



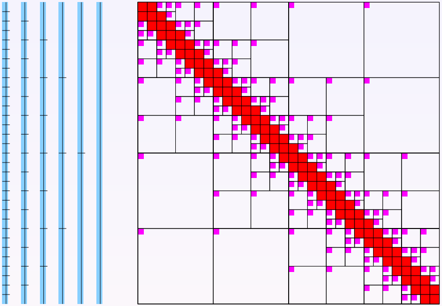
Block tree constructed by recursive splitting.

Admissible blocks factorized

$$G|_{t\times s} \approx Q_t S_{ts} Q_s^*.$$

$\mathcal{H}^2$-matrix

Idea: Split matrix into submatrices that have numerically low rank.



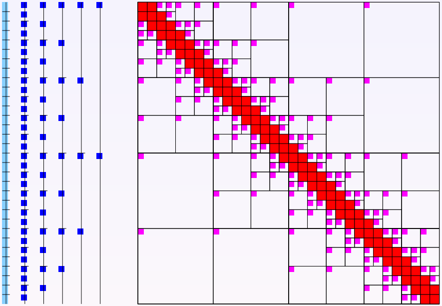
Block tree constructed by recursive splitting.

Admissible blocks factorized

$$G|_{t\times s} \approx Q_t S_{ts} Q_s^*.$$

$\mathcal{H}^2$-matrix

Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

Admissible blocks factorized

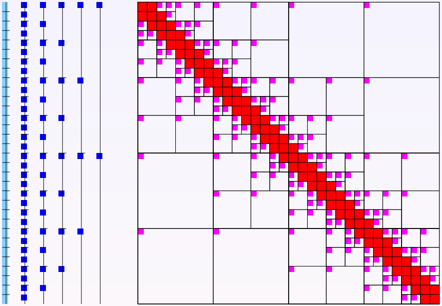
$$G|_{t\times s} \approx Q_t S_{ts} Q_s^*.$$

Cluster basis nested

$$Q_t|_{t'\times k} = Q_{t'} E_{t'}.$$

$\mathcal{H}^2$-matrix

Idea: Split matrix into submatrices that have numerically low rank.



Block tree constructed by recursive splitting.

Admissible blocks factorized

$$G|_{t \times s} \approx Q_t S_{ts} Q_s^*.$$

Cluster basis nested

$$Q_t|_{t' \times k} = Q_{t'} E_{t'}.$$

Result: $\mathcal{O}(nk)$ storage for dense $n \times n$ matrix.

Overview

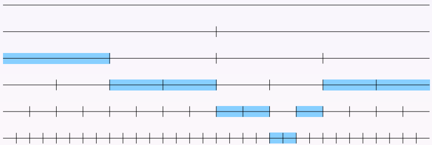
Hierarchical vector

Observation: Solutions of PDEs or IEs are frequently locally smooth.

Goal: Take advantage of smoothness to approximate vectors $x \in \mathbb{R}^n$ by $mk \ll n$ coefficients.

Hierarchical vector: Choose subtree $\mathcal{T}_x$ of cluster tree and use

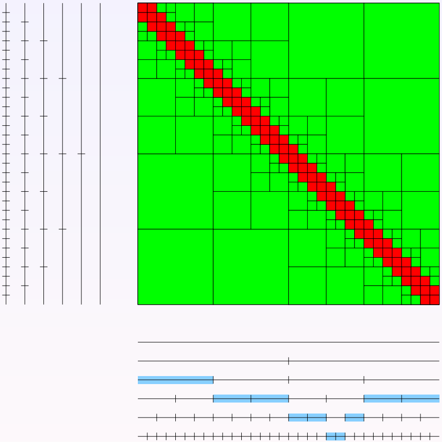
$$x|_t \approx Q_t \hat{x}_t \qquad \text{for all leaves } t \in \mathcal{T}_x.$$



Result: For m leaves, only mk coefficients are required.

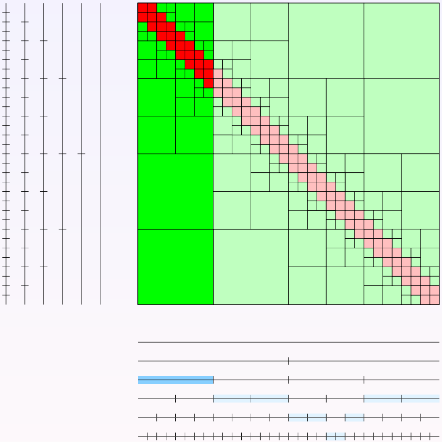
Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.



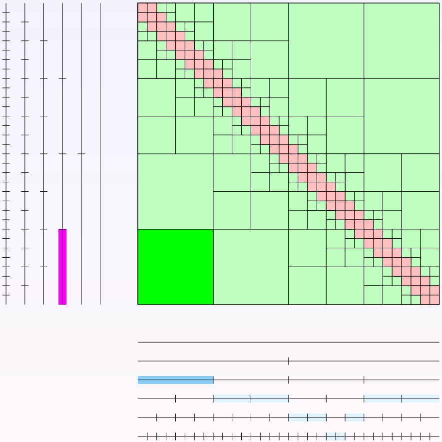
Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.



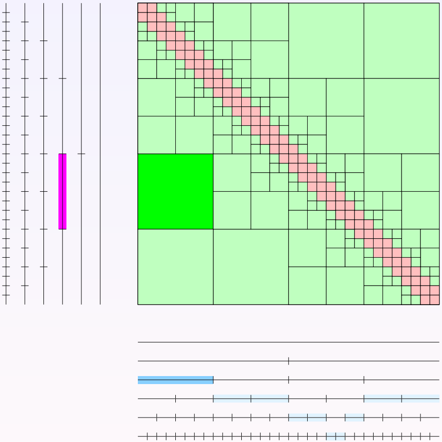
Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.



Admissible blocks already yield
$G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\to$ represented in cluster basis.

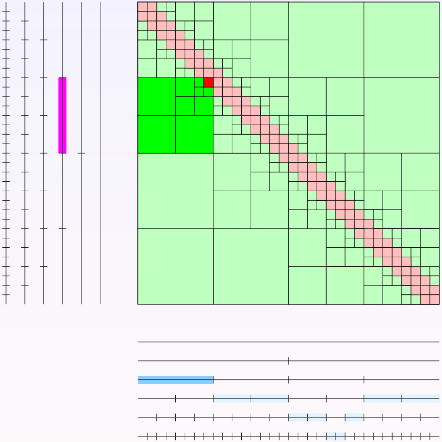
Goal: Efficient matrix-vector multiplication for hierarchical vectors.



Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\to$ represented in cluster basis.

Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.

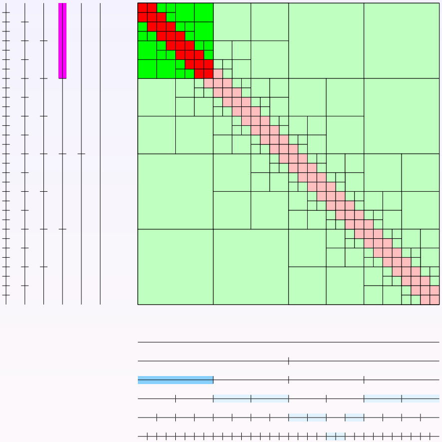


Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\rightarrow$ represented in cluster basis.

Inadmissible blocks handled by extending the basis.

Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.

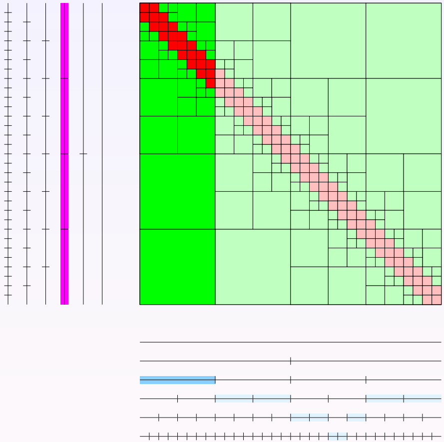


Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\rightarrow$ represented in cluster basis.

Inadmissible blocks handled by extending the basis.

Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.

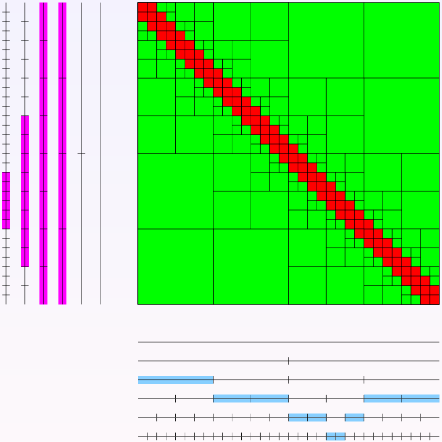


Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\rightarrow$ represented in cluster basis.

Inadmissible blocks handled by extending the basis.

Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.

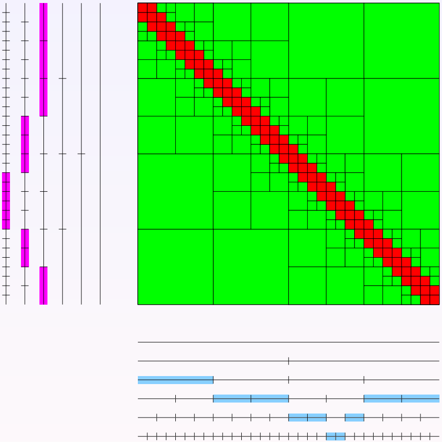


Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\rightarrow$ represented in cluster basis.

Inadmissible blocks handled by extending the basis.

Matrix-vector multiplication

Goal: Efficient matrix-vector multiplication for hierarchical vectors.



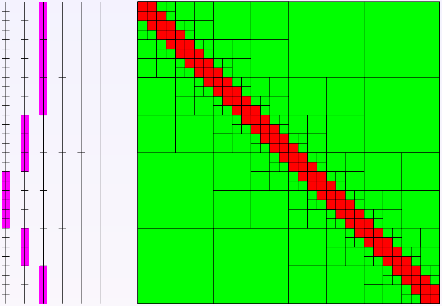
Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\to$ represented in cluster basis.

Inadmissible blocks handled by extending the basis.

Nested structure used to move coefficients to leaves.

Matrix-vector multiplication

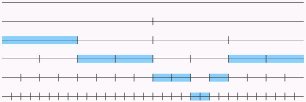
Goal: Efficient matrix-vector multiplication for hierarchical vectors.



Admissible blocks already yield $G|_{t\times s} Q_t \hat{x}_t = Q_t S_{ts} Q_s^* Q_t \hat{x}_t$.
$\rightarrow$ represented in cluster basis.

Inadmissible blocks handled by extending the basis.

Nested structure used to move coefficients to leaves.



Result: $\mathcal{O}(mk)$ operations, product is hierarchical vector.

Induced cluster basis

Problem: How to represent $G|_{t\times s} x|_s = G|_{t\times s} Q_s \hat{x}_s$?

Inadmissible blocks: "Hide" inadmissible submatrices in cluster basis:

$$V_t := \begin{pmatrix} & G|_{t\times s_1} Q_{s_2} & \dots & G|_{t\times s_\ell} Q_{s_\ell} \end{pmatrix},$$

where $G|_{t\times s_1}, \dots, G|_{t\times s_\ell}$ are all inadmissible blocks with row cluster t.
Number of these blocks bounded by sparsity constant c_{sp}.

Induced cluster basis

Problem: How to represent $G|_{t\times s}x|_s = G|_{t\times s}Q_s\hat{x}_s$?

Inadmissible blocks: "Hide" inadmissible submatrices in cluster basis:

$$V_t := \begin{pmatrix} Q_t & G|_{t\times s_1}Q_{s_2} & \dots & G|_{t\times s_\ell}Q_{s_\ell} \end{pmatrix},$$

where $G|_{t\times s_1},\dots,G|_{t\times s_\ell}$ are all inadmissible blocks with row cluster t. Number of these blocks bounded by sparsity constant c_{sp}.

Admissible blocks: Due to $G|_{t\times s}x|_s = Q_tS_{ts}Q_s^*x|_s$, it suffices to include Q_t in the new basis.

Induced cluster basis

Problem: How to represent $G|_{t\times s}x|_s = G|_{t\times s}Q_s\hat{x}_s$?

Inadmissible blocks: “Hide” inadmissible submatrices in cluster basis:

$$V_t := \begin{pmatrix} Q_t & G|_{t\times s_1}Q_{s_2} & \dots & G|_{t\times s_\ell}Q_{s_\ell} \end{pmatrix},$$

where $G|_{t\times s_1},\dots,G|_{t\times s_\ell}$ are all inadmissible blocks with row cluster t. Number of these blocks bounded by sparsity constant c_{sp}.

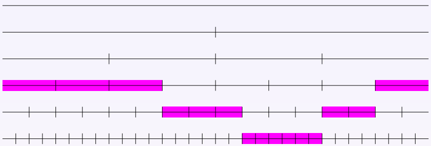
Admissible blocks: Due to $G|_{t\times s}x|_s = Q_t S_{ts} Q_s^* x|_s$, it suffices to include Q_t in the new basis.

Result: Induced cluster basis $(V_t)_t$, inherits nested structure from $(Q_t)_t$. Rank bounded by $(1+c_{\mathrm{sp}})k$, typically significantly larger than k.

Overview

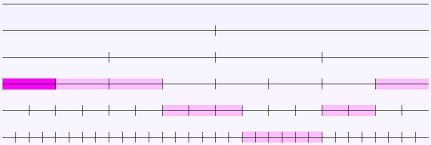
Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



Conversion

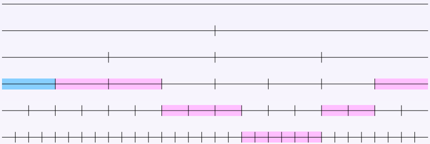
Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



Projection: If resulting error is small, switch back to Q_t.

Conversion

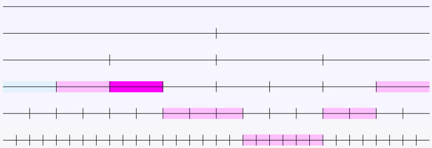
Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



Projection: If resulting error is small, switch back to Q_t.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.

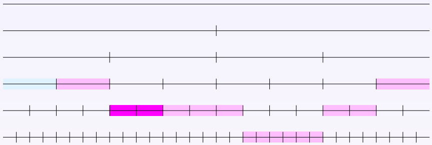


Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.

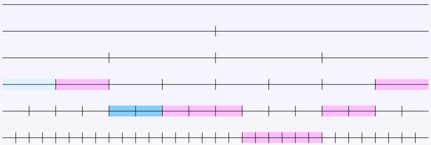


Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.

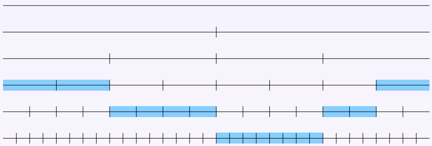


Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.

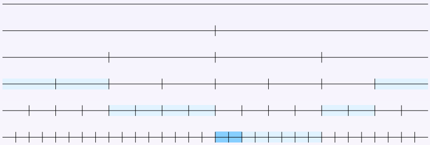


Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



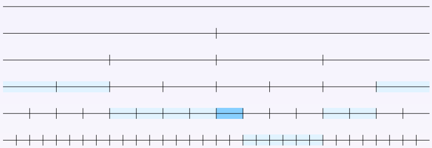
Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Coarsening: If resulting error is small, merge clusters again.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



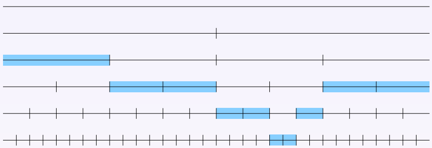
Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Coarsening: If resulting error is small, merge clusters again.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



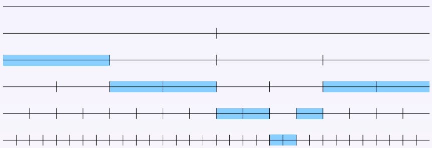
Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Coarsening: If resulting error is small, merge clusters again.

Conversion

Problem: Result $y = Gx$ represented by hierarchical vector with high-rank basis and large number of leaves.



Projection: If resulting error is small, switch back to Q_t.

Refinement: If resulting error is large, split cluster and try again.

Coarsening: If resulting error is small, merge clusters again.

Crucial: We have to be able to control the errors.

Error control

Orthogonal basis: Assuming $Q_t^* Q_t = I$, the best approximation of $y|_t = V_t \hat{y}_t$ in the basis Q_t given by the orthogonal projection

$$Q_t Q_t^* y|_t = Q_t Q_t^* V_t \hat{y}_t.$$

Idea: Extend Q_t to square orthogonal matrix $\begin{pmatrix} Q_t & P_t \end{pmatrix}$.

$$Q_t Q_t^* + P_t P_t^* = \begin{pmatrix} Q_t & P_t \end{pmatrix} \begin{pmatrix} Q_t & P_t \end{pmatrix}^* = I,$$
$$(I - Q_t Q_t^*) V_t = P_t P_t^* V_t.$$

Error matrices: Thin Householder factorization $P_t^* V_t = \widehat{P}_t Z_t$ yields $Z_t \in \mathbb{R}^{k \times k}$ such that

$$\|y|_t - Q_t Q_t^* y|_t\| = \|Z_t \hat{y}_t\|.$$

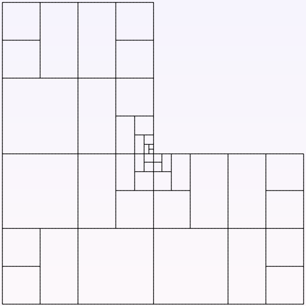
Complexity: Error matrices $(Z_t)_t$ constructed in $\mathcal{O}(nk^2)$ operations.

Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^8 \times 10^{-5}$ with respect to Euclidean norm.

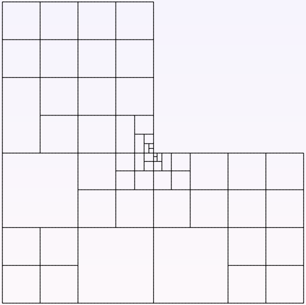


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^7 \times 10^{-5}$ with respect to Euclidean norm.

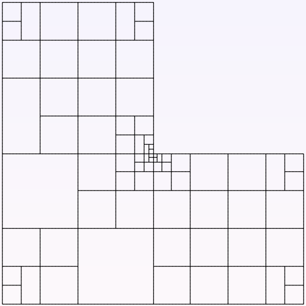


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^6 \times 10^{-5}$ with respect to Euclidean norm.

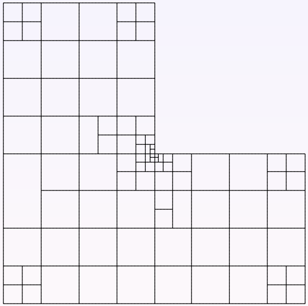


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^5 \times 10^{-5}$ with respect to Euclidean norm.

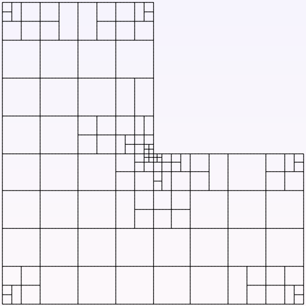


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^4 \times 10^{-5}$ with respect to Euclidean norm.

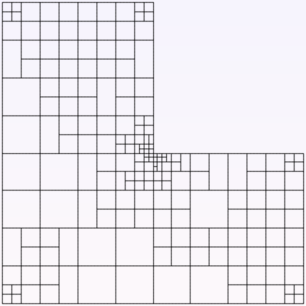


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^3 \times 10^{-5}$ with respect to Euclidean norm.

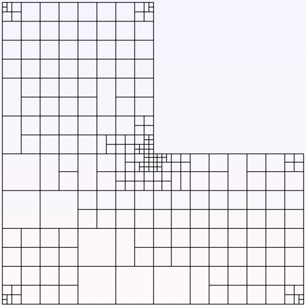


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^2 \times 10^{-5}$ with respect to Euclidean norm.

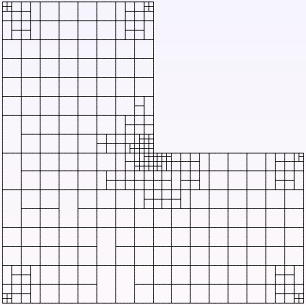


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

Truncation accuracy: $2^1 \times 10^{-5}$ with respect to Euclidean norm.

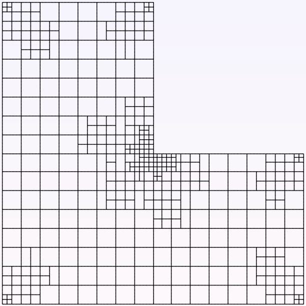


Numerical example

Model problem: L-shaped domain, Poisson equation $-\Delta u = 1$ with homogeneous boundary conditions.

Cluster basis: Bicubic polynomials, regular cluster tree.

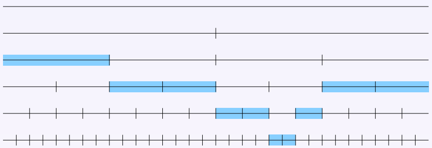
Truncation accuracy: 10^{-5} with respect to Euclidean norm.



Result: Refinement towards singularities, as expected.

Summary

Idea: Represent subvectors in cluster basis, $x|_t = Q_t \hat{x}_t$.
$\to$ Complexity $\mathcal{O}(mk)$ with rank k and m clusters.



Matrix-vector multiplication by $\mathcal{H}^2$-matrix in $\mathcal{O}(mk)$ operations.

Adaptive refinement and coarsening controlled by error matrices: all errors can be computed exactly, "best m-term approximation".

Preprint: http://arxiv.org/abs/1506.00222

h2lib.org