$\mathcal{H}^2$-matrix preconditioners

Steffen Börm
(with Knut Reimer)
funded in part by DFG grants BO 3289-4/1

Christian-Albrechts-Universität zu Kiel

Lothar Collatz CSE Workshop, Plön, 28th of March, 2014

Overview

Overview

Model problem

Poisson's equation with discontinuous and anisotropic coefficients:

$$\begin{aligned} -\operatorname{div} \sigma(x) \operatorname{grad} u(x) &= f(x) && \text{for all } x \in \Omega, \\ u(x) &= 0 && \text{for all } x \in \partial\Omega. \end{aligned}$$

Discretization by finite element or finite difference scheme leads to linear system

$$Ax = b \qquad \text{with } A \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}},\ b \in \mathbb{R}^{\mathcal{I}}.$$

Problem: Differential operator is unbounded.
$\rightarrow$ Condition number of A grows too quickly.

Possible solution: Find preconditioner $C \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ to reduce the condition number, solve

$$CAx = Cb.$$

Ideal preconditioner

Inverse matrix $C := A^{-1}$ would reduce the condition number to one.

Problem: Computing A^{-1} directly

- takes too long and
- requires too much storage.

Approach: Find an approximation of A^{-1}.

Even better: Find an approximation of an LR or Cholesky factorization, evaluate preconditioner by forward and backward substitution.

Properties of the inverse

Positivity: Even for simple examples, we have $(A^{-1})_{ij} > 0$ for all $i, j \in \mathcal{I}$.
$\rightarrow$ Cannot use sparse representation. Not even as an approximation.

Diffusion: For infinite time, the solution y of the vector-valued ODE

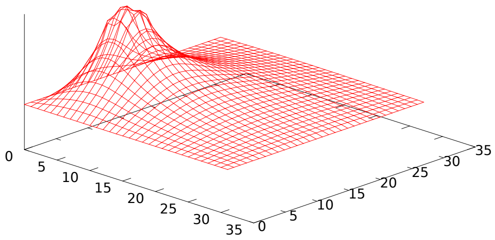
$$y'(t) = b - Ay(t) \qquad \text{for all } t \in \mathbb{R}$$

converges to x, i.e., $\lim_{t\to\infty} y(t) = x = A^{-1}b$.
In the continuous setting: limit of a diffusion process.

Smoothness: Diffusion processes tend to smoothe the solution, at least in the absence of driving forces.

Example: Locally smooth solution

Model problem: $\sigma = 1$, b locally supported.



Observation: Solution increasingly smooth outside of the support of b.

Generalized regularity result

Observation: Let $t, s \subseteq \mathcal{I}$ be subsets of indices that are "geometrically far away" from each other. Then we have

$$\mathrm{supp}(b) \subseteq s \quad \Rightarrow \quad x|_t = (A^{-1}b)|_t \text{ smooth}$$

Idea: Smooth functions can be approximated by polynomials, i.e., in a low-dimensional space V.

$$\mathrm{supp}(b) \subseteq s \quad \Rightarrow \quad x|_t \approx \widetilde{x} \in V.$$

Surprising fact: The latter property also holds for non-smooth and non-isotropic coefficient functions σ.

Result: $A^{-1}|_{t\times s}$ can be approximated by a low-rank matrix as long as the "target cluster" t is far away from the "source cluster" s.

Overview

Factorization

Farfield of a set $t \subseteq \mathcal{I}$ given by

$$\mathrm{far}(t) := \{j \in \mathcal{I} \ : \ \mathrm{dist}(j, t) \geq \mathrm{diam}(t)\},$$

where diameter and distance are suitable geometric quantities.

Generalized regularity results yield that $X|_{t \times \mathrm{far}(t)}$ and $X|_{\mathrm{far}(s) \times s}$ can be approximated by low rank k.

Thin basis matrices $V_t \in \mathbb{R}^{t \times k}$ and $W_s \in \mathbb{R}^{s \times k}$ can be found such that

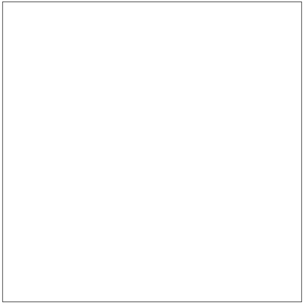
$$X|_{t \times \mathrm{far}(t)} \approx V_t B_t^*, \qquad X|_{\mathrm{far}(s) \times s} \approx A_s W_s^*$$

for suitable matrices B_t, A_s.

Factorization: If $s \subseteq \mathrm{far}(t)$ and $t \subseteq \mathrm{far}(s)$, we have

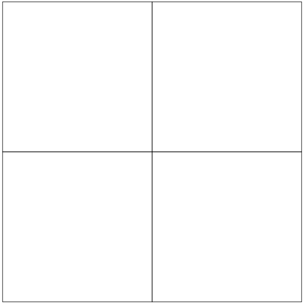
$$X|_{t \times s} \approx V_t S_{ts} W_s^*, \qquad S_{ts} \in \mathbb{R}^{k \times k}.$$

Block structure



Start with entire matrix.

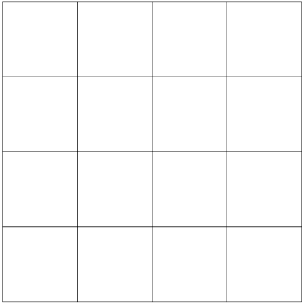
Block structure



Start with entire matrix.

Split into submatrices,

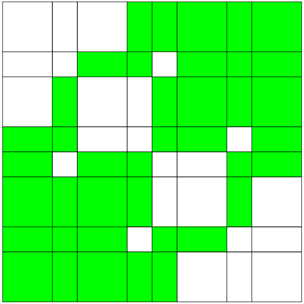
Block structure



Start with entire matrix.

Split into submatrices,

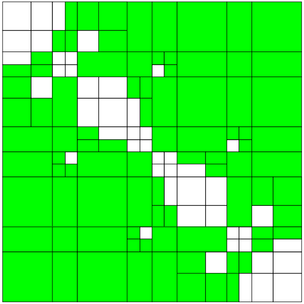
Block structure



Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,

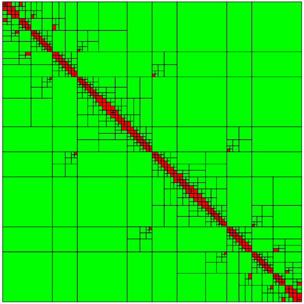
Block structure



Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,

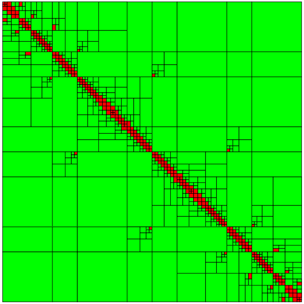
Block structure



Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,
until remaining matrices small enough.

Block structure



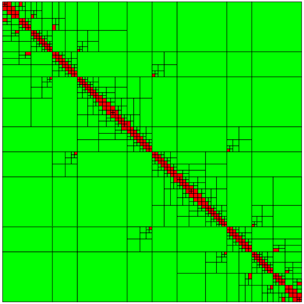
Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,
until remaining matrices small enough.

Result:

- Hierarchy of clusters $t \subseteq \mathcal{I}$.
- Hierarchy of blocks $t \times s \subseteq \mathcal{I} \times \mathcal{I}$.

Block structure



Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,
until remaining matrices small enough.

Result:

- Hierarchy of clusters $t \subseteq \mathcal{I}$.
- Hierarchy of blocks $t \times s \subseteq \mathcal{I} \times \mathcal{I}$.

Matrix representation:

- Farfield blocks in factorized form $X|_{t \times s} \approx V_t S_{ts} W_s^*$.
- Nearfield blocks are small, stored in standard form.
- Cluster bases V_t and W_s in nested form.

$\rightarrow \mathcal{H}^2$-matrix representation, $\mathcal{O}(kn)$ units of storage instead of n^2.

Overview

Goal

We know that A^{-1} can be approximated by an $\mathcal{H}^2$-matrix.

We want to compute this approximation efficiently.

Approach:

- Express A^{-1} in terms of submatrices.
- Take advantage of low-rank factorizations to reduce work.

Block inverse

Block LR factorization yields

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

Denoting the Schur complement by $S := A_{22} - A_{21}A_{11}^{-1}A_{12}$, we find

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

Result: Inverse can be represented by products and inverses of submatrices. The latter can be handled simply by recursion.

Matrix multiplication

Goal: Update $A \leftarrow A + BC$ with $A \in \mathbb{R}^{t \times r}$, $B \in \mathbb{R}^{t \times s}$ and $C \in \mathbb{R}^{s \times r}$.

Recursion applied if B and C are not admissible and subdivided.

$$A \leftarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Matrix multiplication

Goal: Update $A \leftarrow A + BC$ with $A \in \mathbb{R}^{t \times r}$, $B \in \mathbb{R}^{t \times s}$ and $C \in \mathbb{R}^{s \times r}$.

Recursion applied if B and C are not admissible and subdivided.

$$A \leftarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Matrix multiplication

Goal: Update $A \leftarrow A + BC$ with $A \in \mathbb{R}^{t\times r}$, $B \in \mathbb{R}^{t\times s}$ and $C \in \mathbb{R}^{s\times r}$.

Recursion applied if B and C are not admissible and subdivided.

$$A \leftarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Low-rank case: B or C is admissible, therefore given in factorized form.

$$A + BC = A + V_t \; S_{ts} W_s^* C$$

Matrix multiplication

Goal: Update $A \leftarrow A + BC$ with $A \in \mathbb{R}^{t \times r}$, $B \in \mathbb{R}^{t \times s}$ and $C \in \mathbb{R}^{s \times r}$.

Recursion applied if B and C are not admissible and subdivided.

$$A \leftarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Low-rank case: B or C is admissible, therefore given in factorized form.

$$A + BC = A + V_t (S_{ts} W_s^* C)$$

Matrix multiplication

Goal: Update $A \leftarrow A + BC$ with $A \in \mathbb{R}^{t \times r}$, $B \in \mathbb{R}^{t \times s}$ and $C \in \mathbb{R}^{s \times r}$.

Recursion applied if B and C are not admissible and subdivided.

$$A \leftarrow \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Low-rank case: B or C is admissible, therefore given in factorized form.

$$A + BC = A + V_t(S_{ts} W_s^* C) = A + V_t Z^*.$$

Compute $Z = C^* W_s S_{ts}^*$ and perform low-rank update.

Low-rank update

Goal: Update $A \leftarrow A + XY^*$ with $A \in \mathcal{H}^2(V, W)$ and $X, Y \in \mathbb{R}^{\mathcal{I}\times k}$.

Observation: $A + XY^*$ is already an $\mathcal{H}^2$-matrix,
for admissible blocks $t \times s$ we have

$$
\begin{aligned}
(A + XY^*)|_{t\times s} &= V_t S_{ts} W_s^* + X|_{t\times k} Y|_{s\times k}^* \\
&= \underbrace{\begin{pmatrix} V_t & X|_{t\times k} \end{pmatrix}}_{=:\widetilde{V}_t} \underbrace{\begin{pmatrix} S_{ts} & \\ & I \end{pmatrix}}_{=:\widetilde{S}_{ts}} \underbrace{\begin{pmatrix} W_s & Y|_{s\times k} \end{pmatrix}^*}_{=:\widetilde{W}_s^*}.
\end{aligned}
$$

Problem: The rank of $A + XY^*$ increases.

Recompression

Goal: Reduce the rank of $\widetilde{V}_t$ while keeping the resulting error within acceptable bounds.

Tool: Singular value decomposition

$$\widetilde{V}_t = Q \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_k & \\ & & & \ddots \end{pmatrix} P^*$$

with singular values $\sigma_1 \geq \sigma_2 \geq \dots$ and orthogonal Q and P.

Recompression

Goal: Reduce the rank of $\widetilde{V}_t$ while keeping the resulting error within acceptable bounds.

Tool: Singular value decomposition

$$\widetilde{V}_t \approx Q \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_k & \\ & & & 0 \end{pmatrix} P^*$$

with singular values $\sigma_1 \geq \sigma_2 \geq \ldots$ and orthogonal Q and P.
Dropping small singular values yields best approximation.

Recompression

Goal: Reduce the rank of $\widetilde{V}_t$ while keeping the resulting error within acceptable bounds.

Tool: Singular value decomposition

$$\widetilde{V}_t Z_t^* \approx Q \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_k & \\ & & & 0 \end{pmatrix} P^*$$

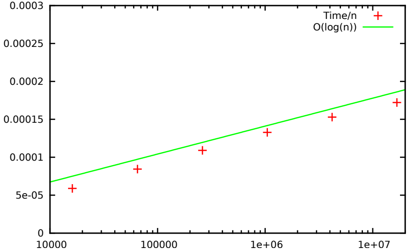
with singular values $\sigma_1 \geq \sigma_2 \geq \ldots$ and orthogonal Q and P.
Dropping small singular values yields best approximation.

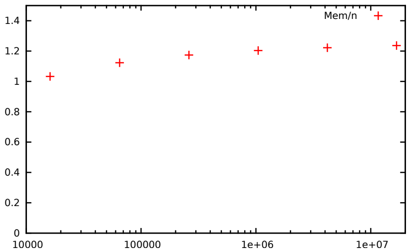
Weight matrices can be used to take the "relative importance" of different columns of $\widetilde{V}_t$ into account.

Overview

Experiment: FEM Cholesky decomposition

Goal: Approximate Cholesky decomposition of a FEM stiffness matrix.



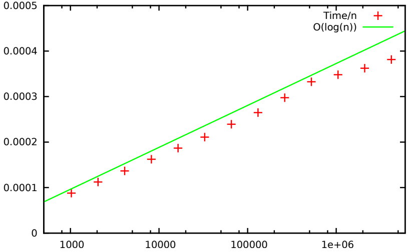


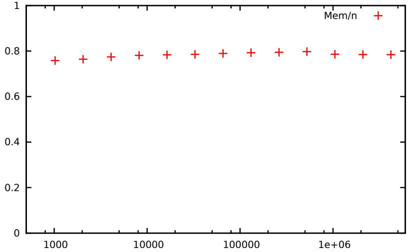
Results:

- Accuracy $\|I - \widetilde{L}^{-*}\widetilde{L}^{-1}A\|_2 \approx 0.1$.
- Factorization in $\sim n \log n$ operations.
- Storage requirements $\sim n$.

Experiment: BEM Cholesky decomposition

Goal: Approximate Cholesky decomposition of a BEM stiffness matrix.





Results:

- Accuracy $\|I - \widetilde{L}^{-*}\widetilde{L}^{-1}A\|_2 \approx 0.2$.
- Factorization in $\sim n \log n$ operations.
- Storage requirements $\sim n$.

Conclusion

Low-rank structure can be used to approximate inverses and factorizations in $\mathcal{O}(n)$.

Efficient algorithms can compute these matrices in $\mathcal{O}(n \log n)$ operations.

H2Lib software package available for scientific research.

Literature:

- L. Grasedyck, W. Hackbusch: Construction and arithmetics of $\mathcal{H}$-matrices, Computing 70:295–334 (2003)
- S. Börm, K. Reimer: Efficient arithmetic operations for rank-structured matrices based on hierarchical low-rank updates, arXiv 1402.5056 (2014)