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

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

Christian-Albrechts-Universität zu Kiel

Workshop on Fast PDE Solvers, Oberwolfach, 16th of May, 2014

Overview

Overview

Model problem

Darcy's equation for groundwater flow leads to

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

with discontinuous, possibly anisotropic, permeabilities $K > 0$.

Discretization by finite element scheme leads to linear system

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

Usually it is ill-conditioned, i.e., not suitable for Krylov solvers.

Goal: Find preconditioner $C \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ to reduce condition number of A, then solve

$$CAx = Cb \qquad \text{or} \qquad C^{1/2}AC^{1/2}\widehat{x} = C^{1/2}b.$$

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

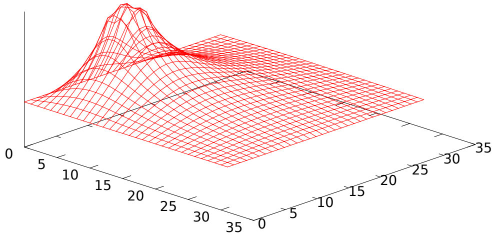
$$y'(t) = b - Ay(t) \qquad \text{for all } t \in \mathbb{R}$$

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: $K = 1$, 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 $\mathcal{V}$.

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

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 $\mathcal{V}$.

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

Generalization: This property also holds (with a different space $\mathcal{V}$) for non-smooth and anisotropic coefficient functions K.

Hackbusch/Bebendorf (2003), B. (2010), Faustmann/Melenk/Praetorius (2013)

Low-rank matrices

Generalized regularity: If $t, s \subseteq \mathcal{I}$ are "far away" from each other, we can find a subspace $\mathcal{V} \subseteq \mathbb{R}^t$ of low dimension k such that

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

Projection: Choose orthogonal $V \in \mathbb{R}^{t \times k}$ with $\operatorname{range}(V) = \mathcal{V}$, then VV^* is the orthogonal projection into $\mathcal{V}$ and we have

$$\operatorname{supp}(b) \subseteq s \quad \Rightarrow \qquad x|_t \approx VV^* x|_t$$

Low-rank matrices

Generalized regularity: If $t, s \subseteq \mathcal{I}$ are "far away" from each other, we can find a subspace $\mathcal{V} \subseteq \mathbb{R}^t$ of low dimension k such that

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

Projection: Choose orthogonal $V \in \mathbb{R}^{t \times k}$ with $\operatorname{range}(V) = \mathcal{V}$, then VV^* is the orthogonal projection into $\mathcal{V}$ and we have

$$\operatorname{supp}(b) \subseteq s \quad \Rightarrow \quad A^{-1}|_{t \times s} b|_s = x|_t \approx VV^* x|_t$$

Low-rank matrices

Generalized regularity: If $t, s \subseteq \mathcal{I}$ are "far away" from each other, we can find a subspace $\mathcal{V} \subseteq \mathbb{R}^t$ of low dimension k such that

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

Projection: Choose orthogonal $V \in \mathbb{R}^{t \times k}$ with $\mathrm{range}(V) = \mathcal{V}$, then VV^* is the orthogonal projection into $\mathcal{V}$ and we have

$$\mathrm{supp}(b) \subseteq s \quad \Rightarrow \quad A^{-1}|_{t\times s} b|_s = x|_t \approx VV^* x|_t = VV^* A^{-1}|_{t\times s} b|_s.$$

Low-rank matrices

Generalized regularity: If $t, s \subseteq \mathcal{I}$ are "far away" from each other, we can find a subspace $\mathcal{V} \subseteq \mathbb{R}^t$ of low dimension k such that

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

Projection: Choose orthogonal $V \in \mathbb{R}^{t \times k}$ with $\operatorname{range}(V) = \mathcal{V}$, then VV^* is the orthogonal projection into $\mathcal{V}$ and we have

$$\operatorname{supp}(b) \subseteq s \quad \Rightarrow \quad A^{-1}|_{t\times s} b|_s = x|_t \approx VV^* x|_t = VV^* A^{-1}|_{t\times s} b|_s.$$

Result: $A^{-1}|_{t\times s} \approx V(V^* A^{-1}|_{t\times s})$, rank-$k$ approximation.

Overview

Factorization

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

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

Generalized regularity: For $t, s \subseteq \mathcal{I}$ we find orthogonal V_t, W_s such that

$$A^{-1}|_{t \times \mathrm{far}(t)} \approx V_t V_t^* A^{-1}|_{t \times \mathrm{far}(t)}, \qquad A^{-1}|_{\mathrm{far}(s) \times s} \approx A^{-1}|_{\mathrm{far}(s) \times s} W_s W_s^*.$$

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

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

Factorization

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

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

Generalized regularity: For $t, s \subseteq \mathcal{I}$ we find orthogonal V_t, W_s such that

$$A^{-1}|_{t \times \mathrm{far}(t)} \approx V_t V_t^* A^{-1}|_{t \times \mathrm{far}(t)}, \qquad A^{-1}|_{\mathrm{far}(s) \times s} \approx A^{-1}|_{\mathrm{far}(s) \times s} W_s W_s^*.$$

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

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

Factorization

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

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

Generalized regularity: For $t, s \subseteq \mathcal{I}$ we find orthogonal V_t, W_s such that

$$A^{-1}|_{t \times \mathrm{far}(t)} \approx V_t V_t^* A^{-1}|_{t \times \mathrm{far}(t)}, \qquad A^{-1}|_{\mathrm{far}(s) \times s} \approx A^{-1}|_{\mathrm{far}(s) \times s} W_s W_s^*.$$

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

$$A^{-1}|_{t \times s} \approx V_t V_t^* A^{-1}|_{t \times s} W_s W_s^* = V_t S_{ts} W_s^*, \qquad S_{ts} \in \mathbb{R}^{k \times k}.$$

Nested bases

Generalized regularity: Consider $t' \subseteq t \subseteq \mathcal{I}$.
Let orthogonal matrices $V_{t'}$, V_t be given as before. We have

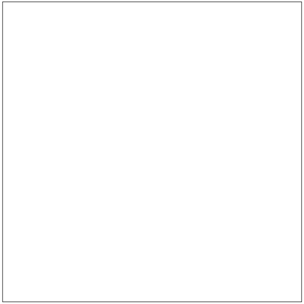
$$A^{-1}|_{t\times \mathrm{far}(t)} \approx V_t V_t^* A^{-1}|_{t\times \mathrm{far}(t)},$$
$$A^{-1}|_{t'\times \mathrm{far}(t')} \approx V_{t'} V_{t'}^* A^{-1}|_{t\times \mathrm{far}(t')}.$$

Restrict to t' and use $\mathrm{far}(t) \subseteq \mathrm{far}(t')$ to get

$$A^{-1}|_{t'\times \mathrm{far}(t)} \approx V_t|_{t'\times k} V_t^* A^{-1}|_{t\times \mathrm{far}(t)},$$
$$A^{-1}|_{t'\times \mathrm{far}(t)} \approx V_{t'} V_{t'}^* A^{-1}|_{t'\times \mathrm{far}(t)}.$$

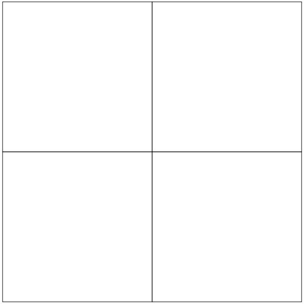
Result: Assume $V_t|_{t'\times k} = V_{t'} E_{t',t}$ with transfer matrix $E_{t',t} \in \mathbb{R}^{k\times k}$.

Block structure



Start with entire matrix.

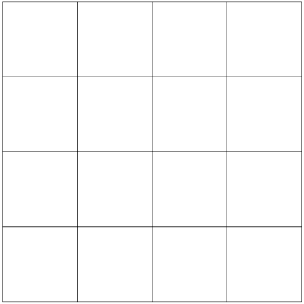
Block structure



Start with entire matrix.

Split into submatrices,

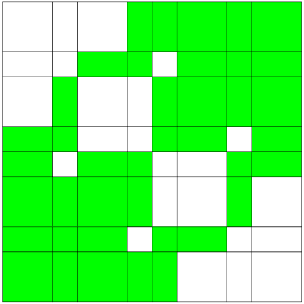
Block structure



Start with entire matrix.

Split into submatrices,

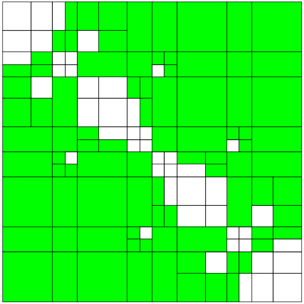
Block structure



Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,

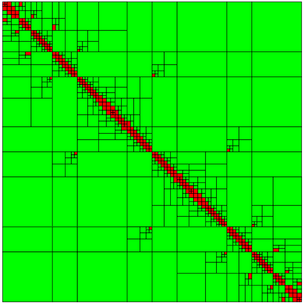
Block structure



Start with entire matrix.

Split into submatrices,
keeping admissible submatrices,

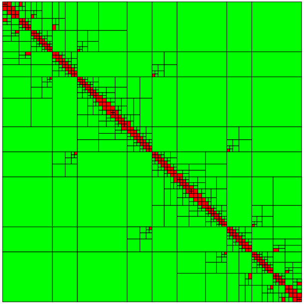
Block structure



Start with entire matrix.

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

Block structure



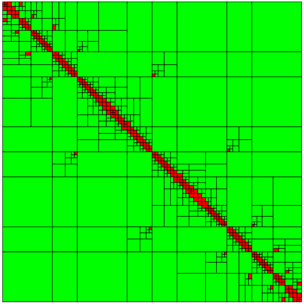
Start with entire matrix.

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 $A^{-1}|_{t \times s} \approx V_t S_{ts} W_s^*$.
- Nearfield blocks are small, stored in standard form.
- Cluster bases V_t and W_s in nested form.

$\rightarrow \mathcal{H}^2$-matrix representation, $\mathcal{O}(nk)$ 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 represented by products and inverses of submatrices.
Inverses of submatrices can be handled by simple recursion.
$\rightarrow$ We need an efficient matrix multiplication algorithm.

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.

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

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.

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

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.

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

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.

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

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.

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

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

$$\widetilde{V}_t = \sum_{\nu=1}^{2k} q_\nu \sigma_\nu p_\nu^*,$$

with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{2k} \geq 0$ and orthonormal bases $\{q_1, \ldots, q_{2k}\}$ and $\{p_1, \ldots, p_{2k}\}$.

Recompression

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

Tool: Singular value decomposition

$$\widetilde{V}_t = \sum_{\nu=1}^{2k} q_\nu \sigma_\nu p_\nu^*, \qquad \widetilde{V}_t \approx \sum_{\nu=1}^{k} q_\nu \sigma_\nu p_\nu^*$$

with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{2k} \geq 0$ and orthonormal bases $\{q_1, \ldots, q_{2k}\}$ and $\{p_1, \ldots, p_{2k}\}$.
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

$$\widetilde{V}_t \widetilde{B}_t = \sum_{\nu=1}^{2k} q_\nu \sigma_\nu p_\nu^*, \qquad \widetilde{V}_t \widetilde{B}_t \approx \sum_{\nu=1}^{k} q_\nu \sigma_\nu p_\nu^*$$

with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{2k} \geq 0$ and orthonormal bases $\{q_1, \ldots, q_{2k}\}$ and $\{p_1, \ldots, p_{2k}\}$.
Dropping small singular values yields best approximation.

Weight matrices $\widetilde{B}_t$ used to implement error control.

Result: Low-rank update in $\mathcal{O}(k^2(\#t + \#s))$ operations.

Summary

Matrix inversion and matrix multiplication can be expressed by

- matrix-vector multiplication of k vectors by submatrix $X|_{t\times s}$,
- low-rank updates $X|_{t\times s} \leftarrow X|_{t\times s} + YZ^*$,

and simple recursion.

Both elementary operations take $\mathcal{O}(k^2(\#t + \#s))$ operations, leading to a total complexity of $\mathcal{O}(nk^2 \log n)$.

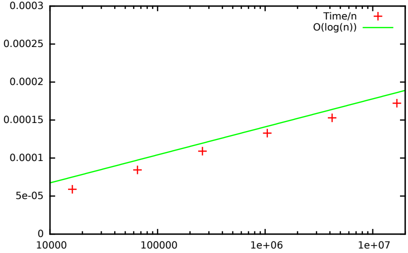
Other algebraic operations can be treated similarly, e.g.,

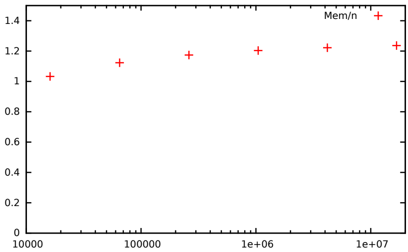
- LR factorization,
- LDL^* and Cholesky factorization,
- matrix exponential.

Grasedyck/Hackbusch (2003), B./Reimer (2014)

Experiment: FEM Cholesky decomposition

Goal: Approximate Cholesky decomposition of a FEM stiffness matrix.



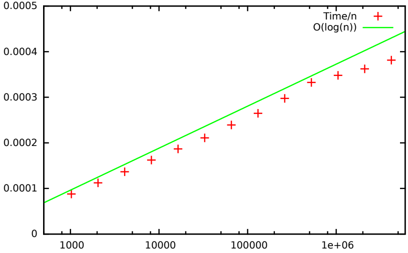


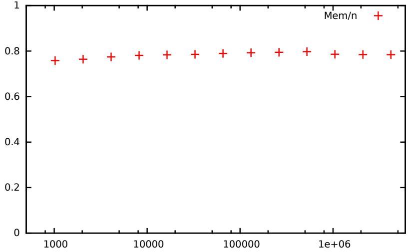
Results:

- 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 $\sim n$.

Overview

Eigenvalue problems

Goal: Compute *m*-th eigenvalue of a symmetric matrix *A*.

Approach: Slicing the spectrum.

1. Start with an interval $[\alpha, \beta]$ containing the eigenvalue.
2. Check how many eigenvalues are less than $\gamma := (\alpha + \beta)/2$.
3. Repeat for $[\alpha, \gamma]$ or $[\gamma, \beta]$.

Parlett (1980)

Eigenvalue problems

Goal: Compute *m*-th eigenvalue of a symmetric matrix *A*.

Approach: Slicing the spectrum.

1. Start with an interval $[\alpha, \beta]$ containing the eigenvalue.
2. Check how many eigenvalues are less than $\gamma := (\alpha + \beta)/2$.
3. Repeat for $[\alpha, \gamma]$ or $[\gamma, \beta]$.

Sylvester's inertia theorem: Use decomposition

$$LDL^* \approx A - \gamma I$$

to find number of eigenvalues less than γ.

Parlett (1980)

Eigenvalue problems

Goal: Compute m-th eigenvalue of a symmetric matrix A.

Approach: Slicing the spectrum.

1. Start with an interval $[\alpha, \beta]$ containing the eigenvalue.
2. Check how many eigenvalues are less than $\gamma := (\alpha + \beta)/2$.
3. Repeat for $[\alpha, \gamma]$ or $[\gamma, \beta]$.

Sylvester's inertia theorem: Use decomposition

$$LDL^* \approx A - \gamma I$$

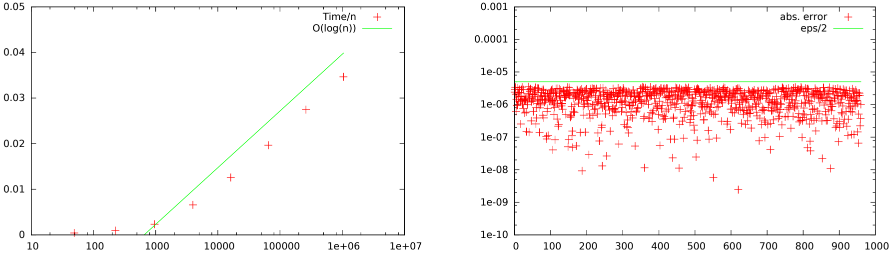
to find number of eigenvalues less than γ.

Higher efficiency: Replace exact factorization by $\mathcal{H}^2$-approximation computed in $\mathcal{O}(n \log n)$ operations.

Parlett (1980), Benner/B./Mach/Reimer (2014)

Experiment: FEM eigenvalues

Goal: Approximate eigenvalues of FEM stiffness matrix.



Results:

- One slice takes $\sim n \log n$ operations.
- Prescribed accuracy guaranteed for all eigenvalues.

Darcy's equation

Goal: Simulate groundwater flow based on Darcy's law.

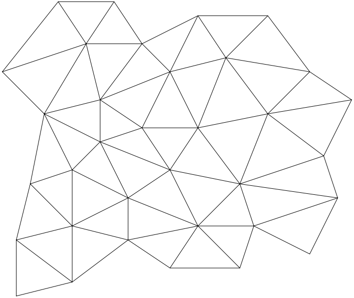
$$\begin{aligned} K(x)^{-1} f(x) + \nabla p(x) &= 0 && \text{for all } x \in \Omega, \\ -\nabla \cdot f(x) &= 0 && \text{for all } x \in \Omega, \\ \langle f(x), n(x) \rangle &= g(x) && \text{for all } x \in \partial\Omega. \end{aligned}$$

Darcy's equation

Goal: Simulate groundwater flow based on Darcy's law.

$$\begin{aligned} K(x)^{-1} f(x) + \nabla p(x) &= 0 && \text{for all } x \in \Omega, \\ -\nabla \cdot f(x) &= 0 && \text{for all } x \in \Omega, \\ \langle f(x), n(x) \rangle &= g(x) && \text{for all } x \in \partial\Omega. \end{aligned}$$

Discretization by mixed FEM:



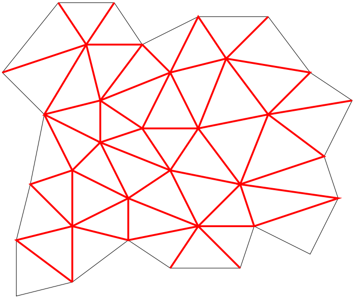
Darcy's equation

Goal: Simulate groundwater flow based on Darcy's law.

$$
\begin{aligned}
K(x)^{-1} f(x) + \nabla p(x) &= 0 && \text{for all } x \in \Omega,\\
-\nabla \cdot f(x) &= 0 && \text{for all } x \in \Omega,\\
\langle f(x), n(x) \rangle &= g(x) && \text{for all } x \in \partial\Omega.
\end{aligned}
$$

Discretization by mixed FEM:

- flux by Raviart-Thomas elements,



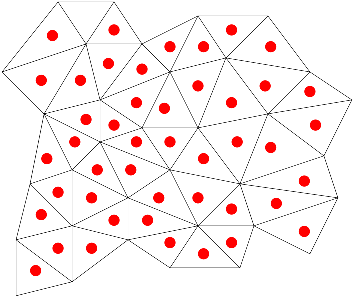
Darcy's equation

Goal: Simulate groundwater flow based on Darcy's law.

$$
\begin{aligned}
K(x)^{-1} f(x) + \nabla p(x) &= 0 && \text{for all } x \in \Omega,\\
-\nabla \cdot f(x) &= 0 && \text{for all } x \in \Omega,\\
\langle f(x), n(x) \rangle &= g(x) && \text{for all } x \in \partial\Omega.
\end{aligned}
$$

Discretization by mixed FEM:

- flux by Raviart-Thomas elements,
- pressure by piecewise constants,



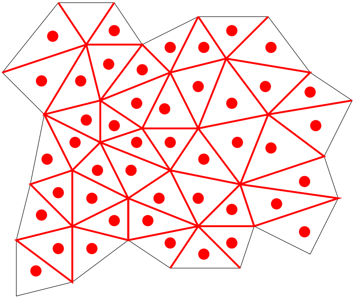
Darcy's equation

Goal: Simulate groundwater flow based on Darcy's law.

$$\begin{aligned} K(x)^{-1} f(x) + \nabla p(x) &= 0 && \text{for all } x \in \Omega, \\ -\nabla \cdot f(x) &= 0 && \text{for all } x \in \Omega, \\ \langle f(x), n(x) \rangle &= g(x) && \text{for all } x \in \partial\Omega. \end{aligned}$$

Discretization by mixed FEM:

- flux by Raviart-Thomas elements,
- pressure by piecewise constants,
- total index set $\mathcal{I} = \mathcal{I}_f \cup \mathcal{I}_p$.



Darcy's equation

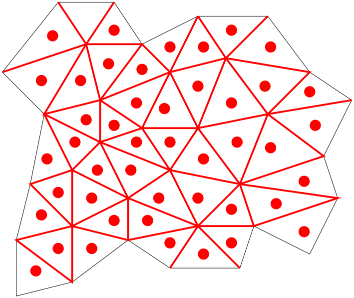
Goal: Simulate groundwater flow based on Darcy's law.

$$
\begin{aligned}
K(x)^{-1} f(x) + \nabla p(x) &= 0 && \text{for all } x \in \Omega,\\
-\nabla \cdot f(x) &= 0 && \text{for all } x \in \Omega,\\
\langle f(x), n(x) \rangle &= g(x) && \text{for all } x \in \partial\Omega.
\end{aligned}
$$

Discretization by mixed FEM:

- flux by Raviart-Thomas elements,
- pressure by piecewise constants,
- total index set $\mathcal{I} = \mathcal{I}_f \cup \mathcal{I}_p$.

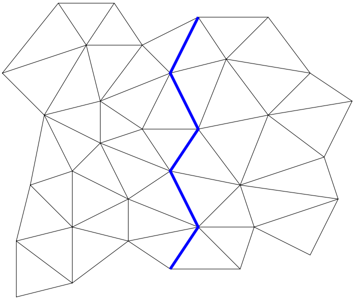
$$
\begin{pmatrix} M & B^* \\ B & \end{pmatrix} \begin{pmatrix} f \\ p \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}
$$



Domain decomposition clustering

Goal: Construct cluster tree ensuring that sub-problems are well-posed Darcy systems.

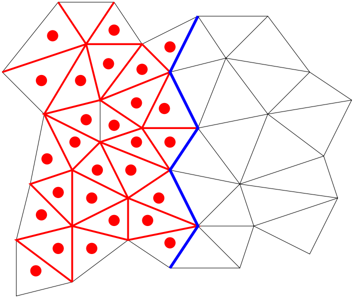
$$\begin{pmatrix} & & \\ & & \\ & & M_{33} \end{pmatrix}$$



Domain decomposition clustering

Goal: Construct cluster tree ensuring that sub-problems are well-posed Darcy systems.

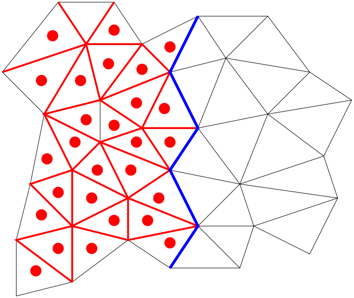
$$\begin{pmatrix} M_{11} & B_{11}^* & & \\ B_{11} & & & \\ & & & \\ & & & M_{33} \end{pmatrix}$$



Domain decomposition clustering

Goal: Construct cluster tree ensuring that sub-problems are well-posed Darcy systems.

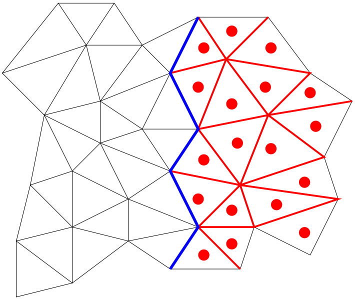
$$\begin{pmatrix} M_{11} & B_{11}^* & & M_{13} \\ B_{11} & & & B_{13} \\ & & & \\ M_{31} & B_{13}^* & & M_{33} \end{pmatrix}$$



Domain decomposition clustering

Goal: Construct cluster tree ensuring that sub-problems are well-posed Darcy systems.

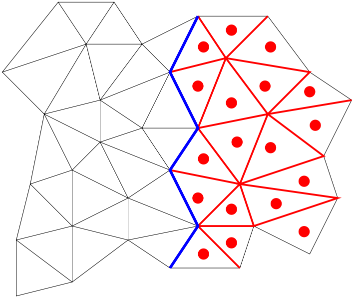
$$\begin{pmatrix} M_{11} & B_{11}^* & & & M_{13} \\ B_{11} & & & & B_{13} \\ & & M_{22} & B_{22}^* & \\ & & B_{22} & & \\ M_{31} & B_{13}^* & & & M_{33} \end{pmatrix}$$



Domain decomposition clustering

Goal: Construct cluster tree ensuring that sub-problems are well-posed Darcy systems.

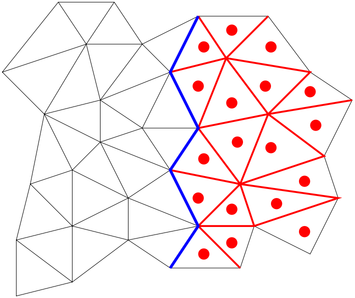
$$\begin{pmatrix} M_{11} & B_{11}^* & & & M_{13} \\ B_{11} & & & & B_{13} \\ & & M_{22} & B_{22}^* & \color{red}{M_{23}} \\ & & B_{22} & & \color{red}{B_{23}} \\ M_{31} & B_{13}^* & \color{red}{M_{32}} & \color{red}{B_{23}^*} & M_{33} \end{pmatrix}$$



Domain decomposition clustering

Goal: Construct cluster tree ensuring that sub-problems are well-posed Darcy systems.

$$\begin{pmatrix} M_{11} & B_{11}^* & & & M_{13} \\ B_{11} & & & & B_{13} \\ & & M_{22} & B_{22}^* & M_{23} \\ & & B_{22} & & B_{23} \\ M_{31} & B_{13}^* & M_{32} & B_{23}^* & M_{33} \end{pmatrix}$$



Clustering strategy: Split cluster t into separator cluster $t_0 \subseteq \mathcal{I}_f \cap t$ and domain clusters $t_1, t_2 \subseteq t$ following domain decomposition approach.

Grasedyck/Kriemann/LeBorne (2009)

Conclusion

$\mathcal{H}^2$-matrices take advantage of low-rank structure to approximate inverses and LR or Cholesky factors in $\mathcal{O}(nk)$ units of storage.

Local low-rank updates allow us to construct approximations in $\mathcal{O}(nk^2 \log n)$ operations.

Applications to preconditioners, symmetric eigenvalue problems and saddle-point problems.

Software package H2Lib scheduled for release in October, 2014.