Iterative Methods for Elliptic Equations with Varying Coefficients

Jan Van lent

BICS
Department of Mathematical Sciences
University of Bath

Computational Science Workshop
Tuesday 10 January 2007

Overview

- elliptic equation with variable coefficient $\alpha \succ 0$

$$\nabla \cdot (\alpha \nabla u) = f$$

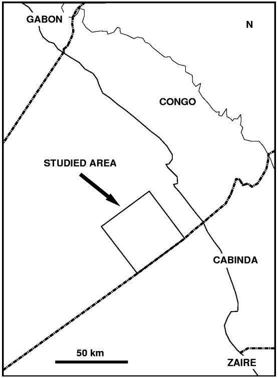
- highly varying α
- motivation : flow in porous media
- finite element discretisation
- large system of equations

$$\mathbf{Au} = \mathbf{f}$$

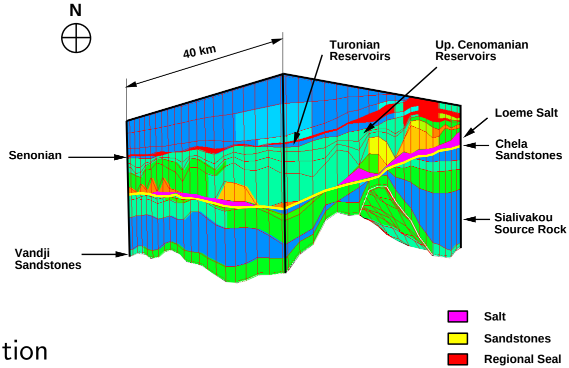
- multilevel iterative methods

Motivation: Sedimentary Basin Simulation

F. Schneider et al., *Oil & Gas Science and Technology* 55(1), 2000

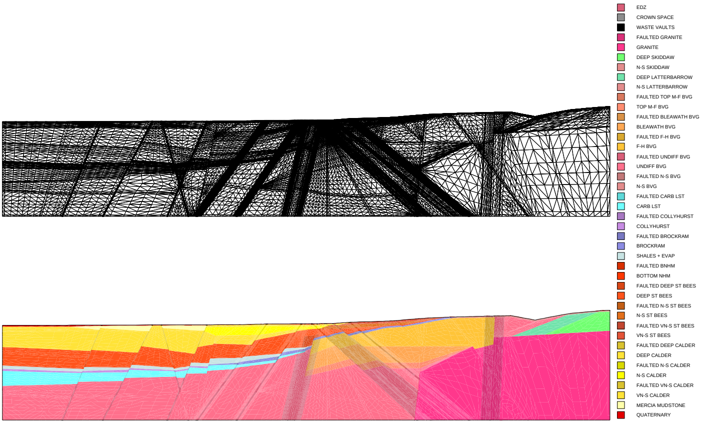


Position of the studied area in the Congo offshore



Li... of the studied 3D-block

Motivation: Groundwater Flow (Sellafield)

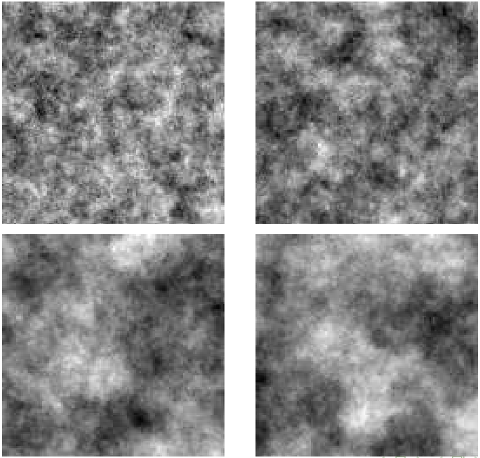


©NIREX UK Ltd.

Motivation: Stochastic Model

Cliffe, Graham, Scheichl, Stals, 2000

- lognormal Gaussian random field
- variance σ^2 : *contrast*
- length scale $\lambda = 5, 10, 20, 50$: *roughness*



Solving the System of Equations

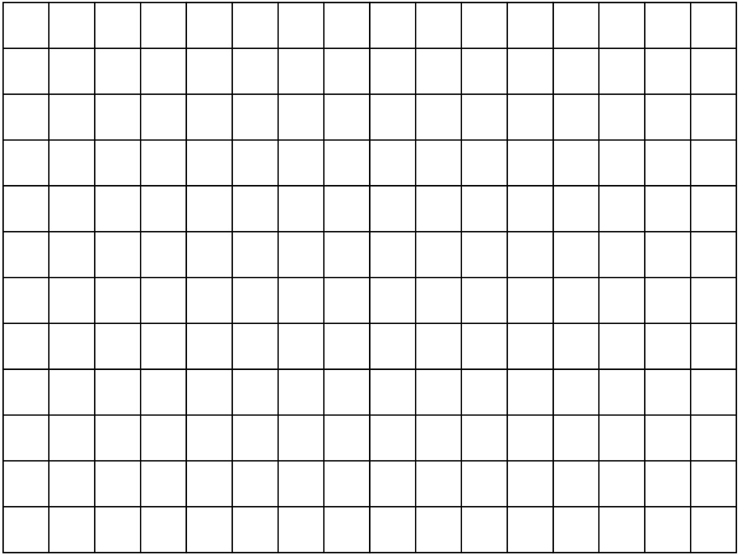
- system of equations
$$Au = f$$
- A is symmetric positive definite
- A is large, but sparse and structured
- 1D, linear elements: tridiagonal
- 2D, regular grid, linear elements:
  block tridiagonal with tridiagonal blocks
- direct solvers for 1D, maybe 2D, not 3D
- constant coefficients: (block-)Toeplitz, FFT
- unstructured grids, varying coefficients:
  multilevel iterative methods

Domain Decomposition Methods

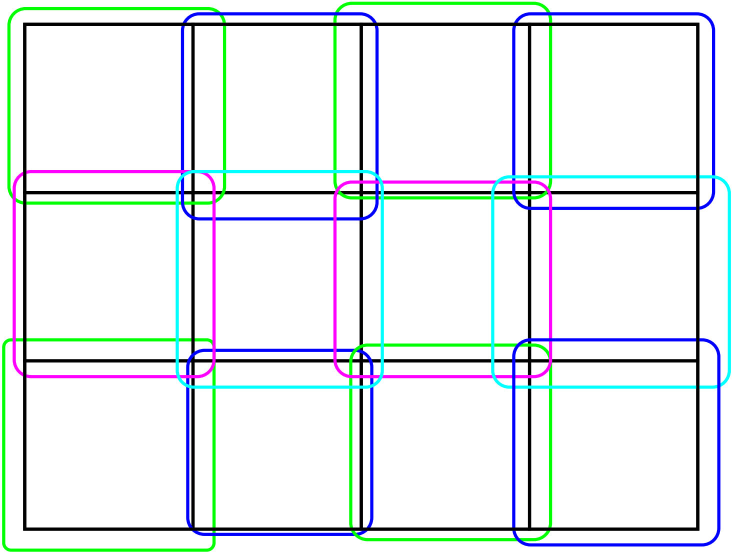
- whole system too much for direct solver (or 1 computer)
- decompose the problem into smaller subproblems
- subproblems are coupled: iteration
- divide domain into smaller subdomains
- many different types
- here overlapping additive Schwarz method

- aim: scalable and robust methods
  number of iterations and cost per iteration well behaved w.r.t.
  - problem size
  - number of subdomains
  - coefficients!
- ideally for N unknowns:
  $O(1)$ iterations, $O(N)$ operations per iteration

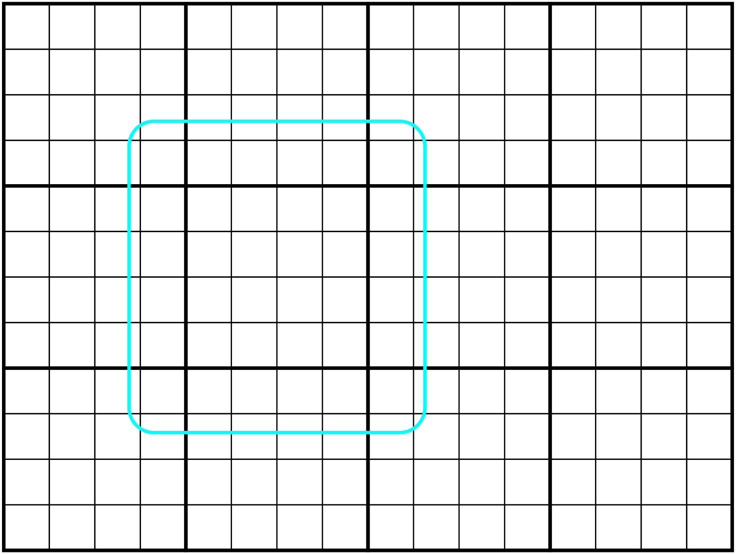
Grid



Overlapping Subdomains

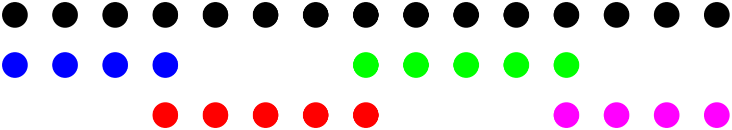


Subdomain



Formulation of the One-Level Method

- CG for A with preconditioner B
- only matrix-vector products for A and B
- number of iterations $\sim \kappa(BA)$
- overlapping subdomains

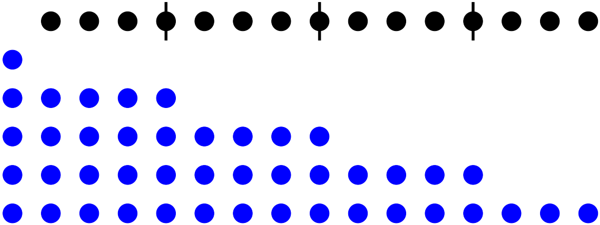


- overlapping additive Schwarz method

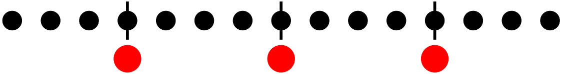
$$
\begin{aligned}
y &= Bx \\
&= \sum_i R_i^T A_i^{-1} R_i x
\end{aligned}
$$

Convergence of the One-Level Method

- not scalable
- illustrate with 1D problem
- rhs $f = 0$, BC $u(0) = 1$, $u(1) = 0$, start with $u^0 = 0$
- information moves at rate of 1 subdomain per iteration



- number of iterations depends on number of subdomains
- remedy: in addition to local solves, do global solve



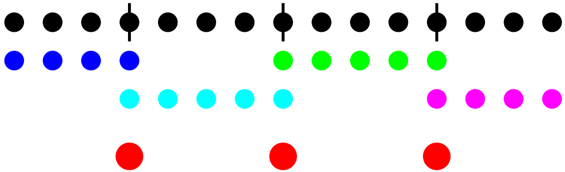
Formulation of the Two-Level Method

- fine level: subproblems that together cover the whole problem

$$B = \sum_i R_i^T A_i^{-1} R_i$$

- coarse level: one smaller problem for the whole domain

$$\hat{B} = R_0^T A_0^{-1} R_0$$



- two-level preconditioner

$$\tilde{B} = \hat{B} + B$$

Choice of Coarse Space

- choice of R_0 is very important for good convergence
- incorporate coefficients
- from theory we know that
  - energy of basis functions must be low
  - basis functions must preserve constants
- set up constrained minimisation problem
- columns of R_0^T

$$R_i^T A_i^{-1} R_i g$$

- where

$$Bg = \mathbf{1}$$

- how to solve this system?

One-Level Preconditioner for the One-Level Preconditioner

- B has special structure, "local" operator
- no global solve needed
- construct one-level preconditioner for B

- matrix A
- one-level preconditioner $B = \sum_i R_i^T A_i^{-1} R_i$
- local problems for $A_i = R_i A R_i^T$
- one-level preconditioner $C = \sum_j R_j^T B_j^{-1} R_j$
- local problems $B_j = R_j B R_j^T$

Efficiency and Robustness

- A_i sparse $\rightarrow$ Bx efficient
- B_i dense
- $B \sim A^{-1}$ and $C \sim B^{-1}$ so somehow $C \sim A$
- Cx can be implemented efficiently
- number of iterations $\sim \kappa(CB)$
- constructing R_0 is scalable and robust

Summary

- considered elliptic equations with varying coefficients
- very large systems of equations
- two-level preconditioner
- construction is not cheap, but algebraic, scalable and robust
- main ideas
  - one-level preconditioner for one-level preconditioner
  - efficient implementation
- topics for further research
  - for overall scalability and robustness,
    it is important to choose the subdomains well
  - non-symmetric systems