Two-Level Preconditioners for Elliptic Equations with Varying Coefficients

Jan Van lent

BICS
Department of Mathematical Sciences
University of Bath

Numerical Analysis Seminar
Friday 15 December 2006

Overview

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

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

- finite element discretisation
- system of equations

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

- preconditioned conjugate gradient
- one-level domain decomposition preconditioner
- two-level domain decomposition preconditioner
- how to construct the second level?

Elliptic Equation with Varying Coefficients (1D)

- 1D elliptic equation
$$(\alpha u_x)_x = f$$
- domain $\Omega = (0, 1)$
- coefficient function $\alpha(x)$
- right hand side $f(x)$
- unknown function $u(x)$
- Dirichlet boundary conditions $u(0) = u(1) = 0$

Elliptic Equation with Varying Coefficients (2D)

- 2D isotropic elliptic equation

$$(\alpha u_x)_x + (\alpha u_y)_y = f$$

- domain $\Omega = (0,1)^2$
- coefficient function $\alpha(x,y)$
- right hand side $f(x,y)$
- unknown function $u(x,y)$
- Dirichlet boundary conditions $u|_{\partial\Omega} = 0$

$$u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0$$

Discretisation of the Continuous Equation

- continuous equation

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

- weak formulation : find u such that

$$a(u, v) = \int \alpha \nabla u \cdot \nabla v = \int fv, \qquad \forall v$$

- finite dimensional approximation : sum of basis functions

$$u \approx \sum_i \mathbf{u}_i \phi_i$$

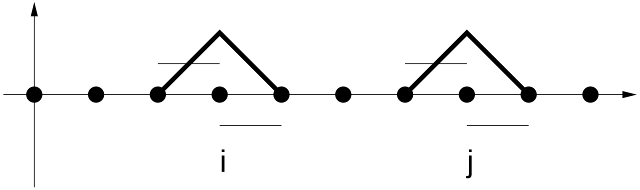
- vector of unknowns $\mathbf{u} = [\mathbf{u}_i]$ is solution of

$$\mathbf{A}\mathbf{u} = \mathbf{f}$$

where $\qquad \mathbf{A} = [a(\phi_i, \phi_j)], \qquad \mathbf{f} = [(f, \phi_i)]$

Finite Element Discretisation (1D)

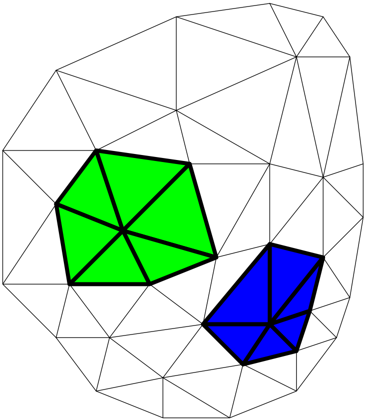
- mesh: subintervals (x_i, x_{i+1})
- piecewise linear approximation
- basis functions



- sparse matrix

Finite Element Discretisation (2D)

- mesh: e.g., triangles
- piecewise linear approximation
- basis functions
- sparse matrix



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

Iterative Methods

- preconditioned Richardson method

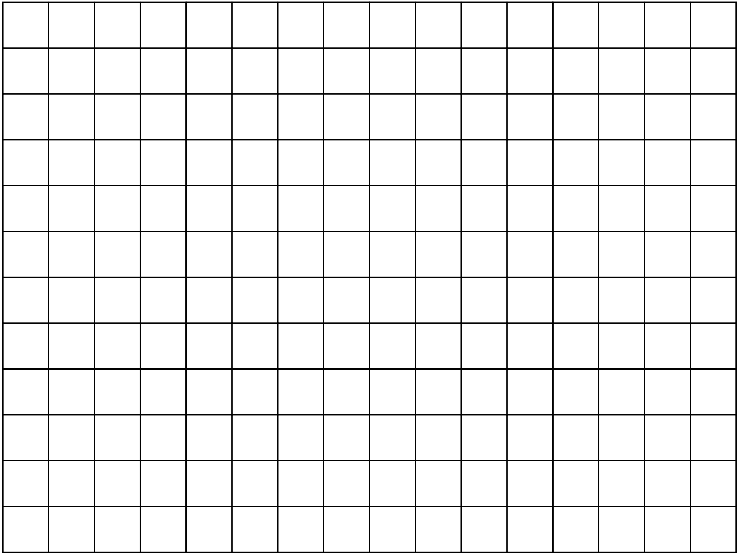
$$u^{k+1} = u^k + B(f - Au^k)$$

- convergence if $\rho(I - BA) < 1$
- preconditioned conjugate gradient method
- only matrix-vector products for A and B
- convergence determined by $\kappa(BA)$
- 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

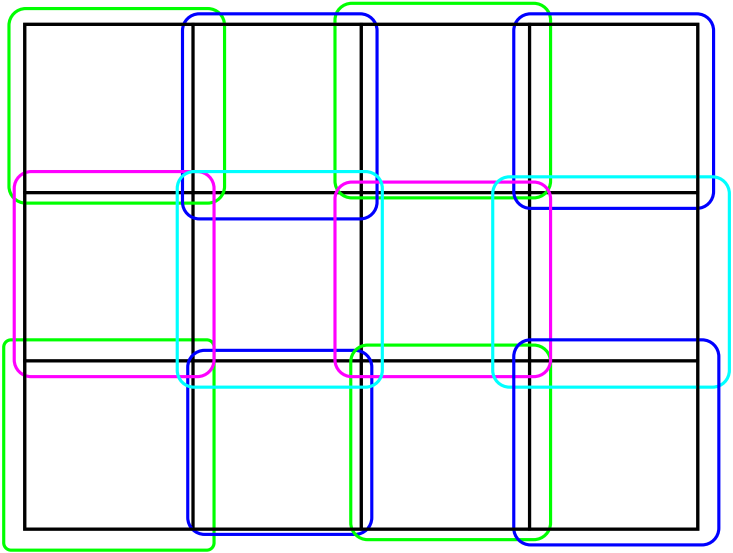
Domain Decomposition Methods

- 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

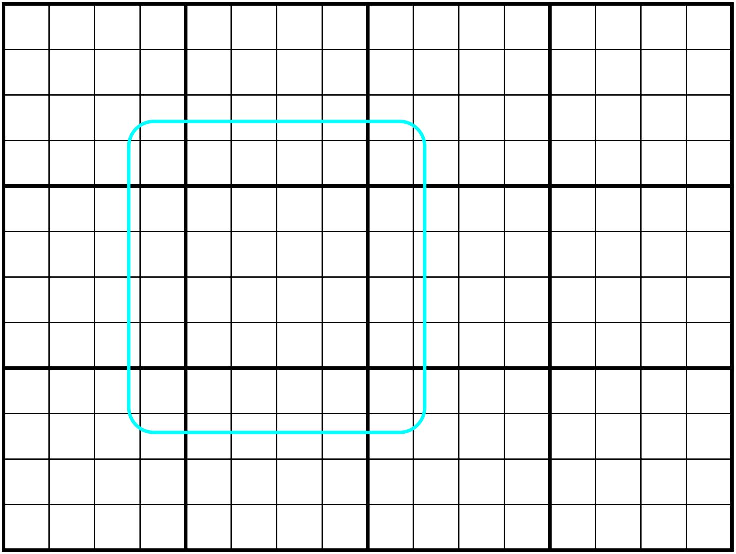
Grid



Overlapping Subdomains

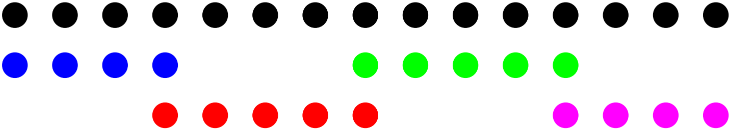


Subdomain



Restriction Matrices (1D)

- overlapping subdomains

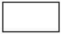
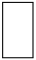
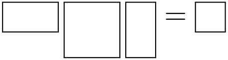


- restriction matrices $R_i = \square$

$$R_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

- extension matrices $R_i^T = \square$

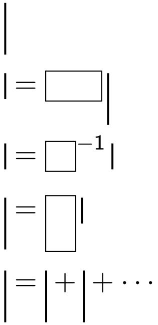
Formulation of the One-Level Method

- restriction of whole space to subspace i : $R_i =$ 
- extension from subspace i into whole space : $R_i^T =$ 
- matrix for subproblem $R_i A R_i^T =$ 
- for injection, A_i is submatrix of A

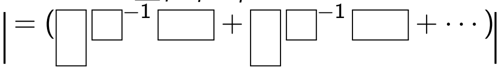
Formulation of the One-Level Method

- overlapping additive Schwarz method

given a vector	x	
restrict to subdomains	$x_i = R_i x$	
solve subproblem	$z_i = A_i^{-1} x_i$	
extend back into whole domain	$y_i = R_i^T z_i$	
sum up all contributions	$y = \sum_i y_i$	

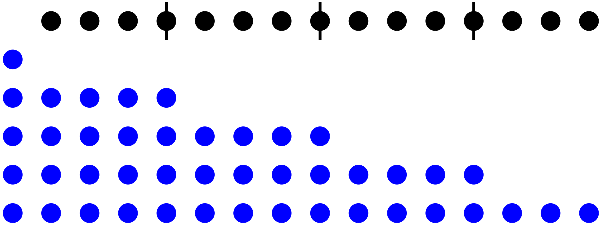


- summary $y = Bx = \sum_i R_i^T A_i^{-1} R_i x$

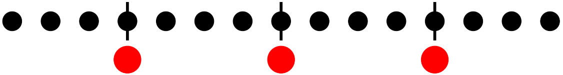


Convergence of the One-Level Method

- 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 1 global solve



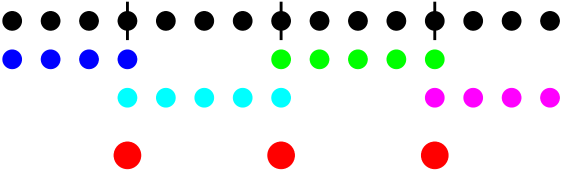
Formulation of the Two-Level Method

- 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

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



- choice of coarse problem
  - one unknown from each subdomain
  - average unknowns in one subdomain
  - weighted average, linear basis functions

Formulation of the Two-Level Method

- system $Au = f$
- restriction matrices R_i
- local problems $A_i = R_i A R_i^T$
- one-level preconditioner

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

- coarse problem $A_0 = R_0 A R_0^T$
- two-level preconditioner

$$\tilde{B} = R_0^T A_0^{-1} R_0 + \sum_i R_i^T A_i^{-1} R_i$$

- rows r_i^T of R_0 contain coefficients of coarse basis functions

Convergence of the Two-Level Method

- choice of coarse space (R_0) is very important for good convergence
- from theory we know that
  - energy of basis functions must be low

$$r_i^T A r_i = \|r_i\|_A^2$$

  - basis functions must preserve constants

$$\sum_i r_i = \mathbf{1}$$

Constructing a Coarse Space Basis

- basis functions defined on subdomains $r_i = R_i^T q_i$
- solution of local problem

$$A_i q_i = g_i \quad (\sim \min_{q_i} \frac{1}{2} q_i^T A_i q_i - q_i^T g_i)$$

- well chosen right hand side g_i
- assume $g_i = R_i g$

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

- preservation of constants $Bg = \mathbf{1}$
- g corresponds to the Lagrange multipliers of a constrained minimisation problem
- how to solve this system?

Preconditioning the One-Level Preconditioner

- precondition B with A

$$\kappa(AB) = \kappa(BA)$$

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

One-Level Preconditioner for the One-Level Preconditioner

- matrix A
- one-level preconditioner $B = \sum_i R_i^T A_i^{-1} R_i$
- local problems for $A_i = R_i A R_i^T$
- A_i is sparse
- one-level preconditioner $C = \sum_j R_j^T B_j^{-1} R_j$
- local problems $B_j = R_j B R_j^T$
- B_j is dense
- $B \sim A^{-1}$ and $C \sim B^{-1}$ so somehow $C \sim A$

Implementing the Preconditioner

- consider a domain j with 2 neighbours k and l

$$R_j R_j^T = I_j, \qquad R_j R_k^T = \hat{I}_{jk} \neq 0, \qquad R_j R_l^T = \hat{I}_{jl} \neq 0$$

- local problem j

$$\begin{aligned} B_j &= R_j(\sum_i R_i^T A_i^{-1} R_i) R_j^T \\ &= A_j^{-1} + \hat{I}_{jk} A_k^{-1} \hat{I}_{kj} + \hat{I}_{jl} A_l^{-1} \hat{I}_{lj} \end{aligned}$$

- all A_i^{-1} are dense
- how can we efficiently apply B_j^{-1}?

Linear Algebra Trick

- local problem solve

$$B_j^{-1} = (A_j^{-1} + \hat{I}_{jk} A_k^{-1} \hat{I}_{kj} + \hat{I}_{jl} A_l^{-1} \hat{I}_{lj})^{-1}$$

- apply Sherman-Morisson-Woodbury formula

$$(A^{-1} + U\Sigma^{-1} V^T)^{-1} = A - AU(\Sigma + V^T AU)^{-1} V^T A$$

- set $\quad A \leftarrow A_j, \quad U = V \leftarrow \begin{bmatrix} \hat{I}_{jk} & \hat{I}_{jl} \end{bmatrix}, \quad \Sigma \leftarrow \begin{bmatrix} A_k & \\ & A_l \end{bmatrix}$
- to give

$$B_j^{-1} = A_j - A_j \begin{bmatrix} \hat{I}_{jk} & \hat{I}_{jl} \end{bmatrix} \left(\begin{bmatrix} A_k & \\ & A_l \end{bmatrix} + \begin{bmatrix} \hat{I}_{kj} \\ \hat{I}_{lj} \end{bmatrix} A_j \begin{bmatrix} \hat{I}_{jk} & \hat{I}_{jl} \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{I}_{kj} \\ \hat{I}_{lj} \end{bmatrix} A_j$$

- factorisation of sparse matrix

Efficiency and Robustness

- number of iterations : $\kappa(CB)$
- cost of C : multiple of cost of B
- constants depend only on
  number of neighbours of subdomains,
  not on number of domains or coefficients
- therefore constructing R_0 is scalable and robust

Related Methods

- same as energy minimising coarse basis functions
  mainly studied for multigrid methods
- multiscale finite elements as coarse basis functions
  different way of choosing boundary conditions of local problems
  for coarse basis functions

Numerical Results for 1D Poisson

n	d	s	$\kappa(A)$	$\kappa(AB)$	$\kappa(DB)$	$\kappa(EB)$	$\kappa(CB)$	$\kappa(\tilde{B}A)$
4	2	2	2e1	5e0	1e1	6e0	2e0	6e0
4	2	4	1e2	2e1	2e1	9e0	3e0	1e1
4	2	8	4e2	9e1	2e1	1e1	3e0	3e1
4	2	16	1e3	4e2	2e1	1e1	3e0	7e1
4	2	32	6e3	1e3	2e1	1e1	3e0	1e2
8	4	2	1e2	8e0	6e1	1e1	2e0	1e1
8	4	4	4e2	4e1	8e1	1e1	3e0	3e1
8	4	8	1e3	1e2	8e1	2e1	4e0	8e1
8	4	16	6e3	8e2	8e1	2e1	4e0	1e2
8	4	32	2e4	3e3	8e1	2e1	4e0	3e2
16	8	2	4e2	1e1	2e2	2e1	3e0	2e1
16	8	4	1e3	8e1	3e2	5e1	4e0	7e1
16	8	8	6e3	3e2	3e2	5e1	5e0	1e2
16	8	16	2e4	1e3	3e2	6e1	5e0	3e2
16	8	32	1e5	6e3	3e2	6e1	5e0	7e2

Numerical Results for 1D Log-Normal Coefficients

n	d	s	$\kappa(A)$	$\kappa(AB)$	$\kappa(DB)$	$\kappa(EB)$	$\kappa(CB)$	$\kappa(\hat{B}A)$
4	2	4	6e5	1e1	2e5	5e2	4e0	1e1
4	2	8	2e10	2e5	1e8	6e7	4e0	9e2
4	2	16	1e8	1e4	5e5	3e5	4e0	1e2
4	2	32	1e10	1e6	1e7	5e6	4e0	2e2
8	4	2	1e6	5e0	1e6	8e3	3e0	1e1
8	4	4	4e7	6e1	7e6	9e3	4e0	7e1
8	4	8	3e7	8e2	1e6	5e5	5e0	1e2
8	4	16	1e12	2e5	1e9	4e7	6e0	3e4
8	4	32	8e11	5e5	2e8	1e5	6e0	1e3
16	8	2	1e5	4e1	2e4	6e2	5e0	7e1
16	8	4	3e9	2e2	1e9	7e5	9e0	3e2
16	8	8	3e9	1e3	4e7	5e5	8e0	9e2
16	8	16	2e12	1e4	5e9	1e7	8e0	3e2
16	8	32	5e12	4e5	4e9	9e6	8e0	1e3

$\alpha_i \in \exp(N(0,4))$

Choice of Subdomains

- construction of the coarse space is robust and scalable
- for given subdomains (R_i), coarse space (R_0) somehow optimal
- iteration for A preconditioned with $\tilde{B}$
- efficiency depends on choice of subdomains
- subdomains should be adapted to coefficients
- current work on aggregation methods

Summary

- considered elliptic equations with varying coefficients
- two-level preconditioner
  for a given set of overlapping subdomains
- construction is not cheap, but algebraic, scalable and robust
- main ideas
  - one-level preconditioner for one-level preconditioner
  - linear algebra trick
- topics for further research
  - analysis of $\kappa(CB)$
  - for overall scalability and robustness,
    it is important to choose the subdomains well
  - cheaper preconditioner for coarse space construction
  - non-symmetric systems

References

- Wan, Chan, Smith, *An Energy-Minimizing Interpolation for Robust Multigrid Methods* (2000)
- Xu, Zikatanov, *On an Energy Minimizing Basis for Algebraic Multigrid Methods* (2004)
- Graham, Lechner, Scheichl, *Domain Decomposition for Multiscale PDEs* (2006)
- Scheichl, Vainikko, *Additive Schwarz with Aggregation-Based Coarsening for Elliptic Problems with Highly Variable Coefficients* (2006)