Domain Decompositionand Projection Methods

C. T. Kelley
NC State University
tim_kelley@ncsu.edu
Research Supported by NSF, DOE, ARO, USACE

DTU ITMAN, 2011

Outline

References

- B. Smith, P. Bjørstad, and W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, Cambridge, 1996.
- Y. Saad, Iterative Methods for Sparse Linear Systems, Prindle, Weber, and Schmidt, New York, 1996.
- J. Mandel, Balancing domain decomposition, Communications in Applied Numerical Methods 9, (1993), pp. 233–241.
- E. W. Jenkins, C. T. Kelley, C. T. Miller, and C. E. Kees, An aggregation-based domain decomposition preconditioner for groundwater flow, SIAM J. Sci. Comp., 23 (2001), pp. 430–441.

Additive Schwarz Preconditioners

Problem: Solve

$$Lu = f \text{ on } \Omega; u = g \text{ on } \Gamma = \partial\Omega.$$

where

- L elliptic (think $L = -\nabla^2$)
- $\Omega = \cup_{i=1}^{p} \Omega_i$

Can we "invert" L on Ω_i and build an effective precondtioner?

Partitioning

- Let A be the matrix representation of a discretization of L
- Suppose R_i is the "restriction" to Ω_i (in the sense of coordinates)
- $A_i = R_i A R_i^T$: restricted operator for the interior grid points in Ω_i
- $B_i = R_i^T A_i^{-1} R_i$
- $T_i = B_i A$

I will put a 1-D example on the board.

One-level Additive Schwarz

The one-level additive Schwarz preconditioner is

$$B = \sum_{i=1}^{p} B_i$$

if the domains are disjoint, this is block-Jacobi.
This only a preconditioner. You cannot build a solver from this.
Note: implict boundary conditions in interior are zero!
There are many other types of Schwarz methods. We will stick to the additive ones.

Two-Level Additive Schwarz

- Ω_C: coarse grid with a few points on each Ω_i
- R_C: restriction map from $\Omega_h \rightarrow \Omega_C$ (full weighting)
- A_C: "coarse grid" operator
- One way is aggregation. Coarse basis are sums of subdomain basis functions.
- $B_0 = R_C^T A_C^{-1} R_C$
- $B = \sum_{i=0}^{p} B_i$

Comments

- The coarse level is necessary to move information between the subdomains.
- Not hard to program on regular girds.
- Very hard to program on unstructured computer-generated grids.
- Ideal preconditioning: $\kappa(BA)$ (or $\kappa(AB)$) independent of h and H.
- You get idea with the Poisson solve preconditioner and with MG.

Notation for Analysis

- h: mesh width for the problem
- H: "size" of the subdomains
- δ: "overlap" of the subdomains
  - Zero overlap: poor preconditioner
  - δ proportional to H: great but expensive
  - $\delta = O(1)$ sometimes optimal

Convergence

- $\kappa(BA) = O(1 + (H/\delta)^2)$.
- $\delta = cH$: $\kappa(BA) = O(1)$ independent of H and h.
- Realistic case: $h < \delta < ch$, so $\kappa(BA) = O(1 + (H/h)^2)$.
- Scalable if you keep H/h fixed as the problem size grows i.e. number of gridpoints/processor is constant.

Projection Methods and DD

Let A be $N \times N$ and

- $S = (1, \dots, N)$
- $S_i \subset S$; $\cup_{i=1}^{p} S_i = S$
- $S_i = (m_i(1), \dots, m_i(n_i))$
- $V_i = [e_{m_i(1)}, \dots, e_{m_i(n_i)}]$ $N \times n_i$
  e_k is coordinate vector.

Here V_i is the subdomain restriction map and we could let

$$A_i = V_i^T A V_i, B_i = V_i^T A_i^{-1} V_i.$$

Additive Projection Method (look familiar?)

for $k = 0, \dots$ **do**
  for $i = 1 : p$ **do**
    Solve $A_i y_i = V_i^T (b - A x_k)$
  end for
  $x_{k+1} = x_k + \sum_{i=1}^{p} y_i$
end for

Residual Reduction

Clearly

$$r_{k+1} = \left(I - \sum_{i=1}^{p} P_i r_k \right)$$

where

$$P_i = AV_i (V_i^T AV_i)^{-1} V_i^T$$

In the very idealized case where the matrices AV_i are orthogonal

$$\sum P_i = I$$

and convergence takes one iteration.