Journal of Computational and Applied Mathematics 261 (2014) 1–13

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Two-level algebraic domain decomposition preconditioners using Jacobi–Schwarz smoother and adaptive coarse grid corrections

Hua Xiang [a], Frédéric Nataf [b]

[a] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, PR China

[b] Laboratoire J.L. Lions, and Alpines-INRIA team, CNRS UMR 7598, Université Pierre et Marie Curie, Paris 75005, France

ARTICLE INFO

Article history:
Received 4 March 2012
Received in revised form 2 October 2013

Keywords:
Domain decomposition method
Jacobi–Schwarz
Ritz vector
Coarse grid correction
Two-level preconditioner

ABSTRACT

We investigate two-level preconditioners on the extended linear system arising from the domain decomposition method. The additive Schwarz method is used as a smoother, and the coarse grid space is constructed by using the Ritz vectors obtained in the Arnoldi process. The coarse grid space can be improved adaptively as the Ritz vectors become a better approximation of the eigenvectors. Numerical tests on the model problem demonstrate the efficiency.

1. Introduction

In many real industrial applications, for example, the solute transport in porous media, the petroleum reservoir simulations, hazardous waste deposition, underground water flow investigations, etc., there arises the following elliptic boundary value problem:

$$\begin{cases} \eta u(\boldsymbol{x}) - \nabla \cdot (\kappa(\boldsymbol{x}) \nabla u(\boldsymbol{x})) = f(\boldsymbol{x}), & \text{in } \Omega, \\ u(\boldsymbol{x}) = g_D(\boldsymbol{x}), & \text{on } \Gamma_D, \\ \boldsymbol{n} \cdot (\kappa(\boldsymbol{x}) \nabla u(\boldsymbol{x})) = g_N(\boldsymbol{x}), & \text{on } \Gamma_N, \end{cases} \tag{1}$$

where κ is the diffusion tensor, g_D and g_N are known functions, Γ_D and Γ_N are the boundaries imposed with the Dirichlet boundary condition and the Neumann boundary condition respectively. The problem is heterogeneous if there are strong variations in κ; The problem is anisotropic if the ratio between any two eigenvalues of κ is large. The heterogeneity and anisotropy may bring forth great difficulties in numerical simulations.

For large scale problem, parallel algorithms are used to solve (1). The domain decomposition method (DDM) is one of the most successful parallel methods. The DDM on (1) will yield a large sparse linear system $Au = f$, where $A \in \mathbb{R}^{n\times n}$ and $f \in \mathbb{R}^n$. The iterative solvers such as Krylov subspace methods may converge very slowly or even stagnate when there exist big discontinuities in κ, especially when the discontinuities lie along the interface. In this paper we will focus on how to accelerate the convergence by applying preconditioners.

One classical preconditioner is the additive Schwarz (AS). Define a matrix R_i to be the restriction matrix for subdomain Ω_i such that $R_i = [0, I_{\Omega_i}, 0]$, where I_{Ω_i} is an identity matrix with the size of subdomain Ω_i. The one-level additive Schwarz

E-mail addresses: hxiang@whu.edu.cn, xiang@ann.jussieu.fr (H. Xiang), nataf@ann.jussieu.fr (F. Nataf).

0377-0427/$ – see front matter
http://dx.doi.org/10.1016/j.cam.2013.10.027

(AS) preconditioner reads

$$M_{AS}^{-1} := \Sigma_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i,$$

where N is the number of subdomains.

But this one-level preconditioner is non-scalable even for the homogeneous case where κ is constant. The dependence of subdomains is global in nature for elliptic problems, whereas the information among the subdomains only exchanges through the overlapping regions or the interfaces. It needs at least $N-1$ iterations when the problem is divided into N subdomains in a one-way partition. To achieve a good scalability, one needs a coarse solver, which transfers certain global information among the subdomains. By adding the coarse problem to the one-level preconditioner, we have the following two-level additive Schwarz preconditioner

$$P_{AS}^{-1} := M_{AS}^{-1} + R_0^T (R_0 A R_0^T)^{-1} R_0, \tag{2}$$

where R_0 is a restriction on to the coarse space. For the case where κ is constant, the condition number of the preconditioned matrix satisfies (cf. [1])

$$\kappa(P_{AS}^{-1} A) \le c \left(1 + \frac{H}{\delta}\right),$$

where H is the size of subdomain, δ is the overlap size, c is a constant independent of H and δ. The overlap can be chosen to be of several mesh sizes. We can see that the algorithm is scalable.

The first term in (2) is a fine grid solver, and can be performed very efficiently in parallel. The second term is the coarse grid correction, which is solved by every processor by using, for example, a dense LU factorization. The BPS preconditioner introduced by Bramble, Pasciak and Schatz in [2] is of this type.

The balancing Neumann–Neumann preconditioner, which is well-known and carefully investigated like the FETI algorithm in the domain decomposition method, has the similar form. For symmetric systems the balancing preconditioner was proposed by Mandel in [3]. The abstract balancing preconditioner for nonsymmetric systems reads (cf. [4])

$$P_{BNN} := Q_D M^{-1} P_D + Z E^{-1} Y^T, \tag{3}$$

where $E = Y^T A Z$, $P_D = I - AZE^{-1}Y^T$, $Q_D = I - ZE^{-1}Y^T A$. For the symmetric positive definite (spd) system, choosing $Y = Z$, the authors in [5] define

$$P_{ADEF2} := Q_D M^{-1} + Z E^{-1} Z^T. \tag{4}$$

These are quite efficient preconditioners. But they can also exhibit slow converge when solving the elliptic boundary value problem (1) with large discontinuities in the diffusion tensor κ, especially when the discontinuities are along the subdomain interfaces. The key point is that we need certain proper coarse grid spaces. In [6] we propose the way to construct the coarse grid spaces based on the local Dirichlet-to-Neumann maps and the harmonic extensions. It is suitable for the parallel implementation, and quite efficient, even for the case with large discontinuities in the coefficients. The parallel performance of the two-level additive Schwarz preconditioner (2) depends on the definition of the coarse space. In this paper we present an adaptive, and purely algebraic method to construct the coarse grid space, together with the two-level preconditioners. Precisely speaking, we use the approximate eigenvectors, such as Ritz vectors, harmonic Ritz vectors, etc., to construct the coarse space.

Our main contribution in this paper is that we use the Ritz vectors to construct the coarse grid space, and present an algorithm based on the framework of FGMRES [7], which combine the Jacobi–Schwarz smoothers and the adaptive coarse grid corrections.

The organization of the paper is as follows. In Section 2 we investigate the two-level preconditioners and their spectral properties. In Section 3 we give a heuristic way to construct the coarse grid space by using the Ritz vectors or the harmonic Ritz vectors. And an adaptive strategy to use the coarse grid space based on the approximate eigenvectors is introduced in Section 4. The model problems partitioned in one-way and two-way are investigated in detail in Section 5.

2. Two-level preconditioners

We discretize the elliptic boundary value problem (1) defined in domain Ω. Without loss of generality, we assume that the domain Ω is decomposed into two overlapping subdomains Ω_1 and Ω_2. The overlapping region is $\Omega_1 \cap \Omega_2$.

Let the subscript I denote the set of indices of the interior points, and the subscript O stand for the indices of the overlapping points. The superscript (i) represents the quantity corresponding to the i-th subdomain Ω_i. At the algebraic level this corresponds to a partition of the set of indices $\mathcal{N}$ into three sets: $\mathcal{N}_I^{(1)}$, $\mathcal{N}_O$ and $\mathcal{N}_I^{(2)}$.

The discretization yields the original linear system of the following form

$$\begin{bmatrix} A_{II}^{(1)} & A_{IO}^{(1)} & \\ A_{OI}^{(1)} & A_{OO} & A_{OI}^{(2)} \\ & A_{IO}^{(2)} & A_{II}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_O \\ u_I^{(2)} \end{bmatrix} = \begin{bmatrix} f_I^{(1)} \\ f_O \\ f_I^{(2)} \end{bmatrix}. \tag{5}$$