Journal of Computational and Applied Mathematics 273 (2015) 274–295

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Preconditioners based on the Alternating-Direction-Implicit algorithm for the 2D steady-state diffusion equation with orthotropic heterogeneous coefficients[☆]

Longfei Gao [a,*], Victor M. Calo [b]

[a] Applied Mathematics & Computational Science, King Abdullah University of Science and Technology, Saudi Arabia

[b] Center for Numerical Porous Media, Applied Mathematics & Computational Science, Earth Science & Engineering, King Abdullah University of Science and Technology, Saudi Arabia

ARTICLE INFO

Article history:
Received 13 December 2013
Received in revised form 26 May 2014

Keywords:
Alternating Direction Implicit
Kronecker product
Finite element method
Tensor product basis functions
Preconditioning
Compound iteration

ABSTRACT

In this paper, we combine the Alternating Direction Implicit (ADI) algorithm with the concept of preconditioning and apply it to linear systems discretized from the 2D steady-state diffusion equations with orthotropic heterogeneous coefficients by the finite element method assuming tensor product basis functions. Specifically, we adopt the compound iteration idea and use ADI iterations as the preconditioner for the outside Krylov subspace method that is used to solve the preconditioned linear system. An efficient algorithm to perform each ADI iteration is crucial to the efficiency of the overall iterative scheme. We exploit the Kronecker product structure in the matrices, inherited from the tensor product basis functions, to achieve high efficiency in each ADI iteration. Meanwhile, in order to reduce the number of Krylov subspace iterations, we incorporate partially the coefficient information into the preconditioner by exploiting the local support property of the finite element basis functions. Numerical results demonstrated the efficiency and quality of the proposed preconditioner.

1. Introduction

The Alternating Direction Implicit (ADI) algorithm, which belongs to the category of matrix-splitting iterative methods, was first proposed almost six decades ago for solving parabolic and elliptic partial differential equations, see [1–4]. With a proper set of acceleration parameters, the ADI algorithm can be very powerful, given that each iteration can be performed efficiently. For example, consider the 2D Poisson equation, defined on a rectangle with full Dirichlet boundary conditions:

$$-\left(u_{,xx}+u_{,yy}\right)=f. \qquad (1)$$

When discretized by the finite element method, see [5–7], with tensor product basis functions, each of $u_{,xx}$ and $u_{,yy}$ leads to a matrix that possesses the Kronecker product structure, see [8–10]. This Kronecker product structure can be exploited to perform each ADI iteration efficiently, see [11].

☆ This work was supported in part by the King Abdullah University of Science and Technology (KAUST) Center for Numerical Porous Media and by an Academic Excellence Alliance program award from KAUST's Global Collaborative Research under the title "Seismic wave focusing for subsurface imaging and enhanced oil recovery".

* Corresponding author. Tel.: +966 544701501.
E-mail addresses: longfei.gao@kaust.edu.sa (L. Gao), victor.calo@kaust.edu.sa (V.M. Calo).

http://dx.doi.org/10.1016/j.cam.2014.06.021

However, this elegant algorithm is fragile and its application is limited to certain model problems only, mainly due to the fact that the Kronecker product structure can be easily destroyed by the appearance of complex coefficients. When encountered with complex coefficients, we resort to the compound iteration idea, originally proposed in [12], to extend the practice of the ADI algorithm. Specifically, we first construct an approximate matrix for which the ADI algorithm can be applied efficiently. Then, a fixed number of ADI iterations is applied to invert this approximate matrix, serving as the preconditioner for the original matrix. These iterations are called the inner iterations. Outside these ADI iterations, another iterative scheme, called the outer iterations, is applied to solve the preconditioned linear system.

The convergence speed of this compound iteration scheme depends on the quality of the approximate matrix. In this paper, we focus on linear systems arising from the finite element discretization with tensor product basis functions of the 2D steady-state diffusion equations with orthotropic heterogeneous coefficients. For these linear systems, the ADI algorithm is not applicable directly due to the appearance of these coefficients. By exploiting the local support property of the basis functions used in the finite element discretization, we are able to construct an approximate matrix that can incorporate partially the coefficient information and meanwhile, enable the use of the ADI algorithm. Various numerical tests are performed to verify the efficiency of the compound iteration scheme with the proposed preconditioner.

The rest of this paper is organized as follows. In Section 2, we briefly describe the ADI algorithm. In Section 3, we present the definition and properties of the Kronecker product, as well as its associated efficient algorithms. In Section 4, we examine the linear system arising from the 2D Poisson equation by the finite element discretization with tensor product basis functions and demonstrate how the Kronecker product structure can be combined with the ADI algorithm to solve this linear system efficiently. In Section 5, we discuss the 2D steady-state diffusion equations with orthotropic heterogeneous coefficients. We present a preconditioner based on the ADI algorithm and the local support property of the basis functions. Numerical examples are presented in Section 6 to demonstrate the performance of this preconditioner. Finally, we conclude with Section 7.

2. Alternating Direction Implicit algorithm

The Alternating Direction Implicit (ADI) algorithm was originally proposed in [1] in 1955 as an iterative method for parabolic and elliptic partial differential equations (PDEs). We consider the linear system:

$$(K^X + K^Y)b = \mathcal{F}, \tag{2}$$

where K^X and K^Y are square matrices while b and $\mathcal{F}$ are vectors of compatible sizes.

The following scheme can be used to solve (2) iteratively:

$$(r^{(k)}\Sigma + K^X)b^{\left(k+\frac{1}{2}\right)} = (r^{(k)}\Sigma - K^Y)b^{(k)} + \mathcal{F}; \tag{3a}$$

$$(r^{(k)}\Sigma + K^Y)b^{(k+1)} = (r^{(k)}\Sigma - K^X)b^{\left(k+\frac{1}{2}\right)} + \mathcal{F}, \tag{3b}$$

where $k = 0, 1, 2, \dots$ stands for the iteration step, Σ is a matrix with the same size of K^X or K^Y, $b^{(k)}$ is the approximation of the solution vector b at iteration step k and $r^{(k)}$ is a scalar number, called the acceleration parameter, to be explained with details in Section 4. If both (3a) and (3b) can be solved efficiently and the total amount of iterations is reasonable, the overall cost of this algorithm can be very low. This is the underlying idea of the ADI algorithm.

At first glance, the assumption that both (3a) and (3b) can be solved efficiently might seem questionable. However, if linear system (2) arises from the finite difference discretization with the five-point stencil of the 2D Poisson equation (1):

$$-\left(u_{,xx} + u_{,yy}\right) = f,$$

where $u_{,xx}$ and $u_{,yy}$ correspond to K^X and K^Y, respectively, then both K^X and K^Y can be reformulated as tridiagonal matrices after suitable permutations.

An efficient algorithm (the tridiagonal matrix algorithm, also known as the Thomas algorithm, see [13,14]) can be applied to invert non-singular tridiagonal matrices efficiently. Therefore, if adding the extra term $r^{(k)}\Sigma$ does not destroy this tridiagonal structure in K^X and K^Y, for instance, when Σ is the identity matrix, (3a) and (3b) can be solved efficiently. This is the original motivation behind the development of this iterative scheme.

However, there are other cases where the individual equations in (3) can be solved efficiently. One noticeable example appears when the Poisson equation (1) is discretized by the finite element method with tensor product basis functions, see [11]. In this case, both K^X and K^Y, which still correspond to $u_{,xx}$ and $u_{,yy}$ in (1), respectively, possess the Kronecker product property. We briefly explain the Kronecker product in Section 3.

The inverse of a non-singular Kronecker product matrix can be applied efficiently, see [15–17]. If in addition, the extra term $r^{(k)}\Sigma$ does not destroy the Kronecker product structure and the invertibility of K^X and K^Y, both (3a) and (3b) can be solved efficiently and therefore, the ADI algorithm is still applicable.