ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 1, 2014, *pp*. 022-030

Conservative domain decomposition procedure for the variable coefficient diffusion equation

Ruiyi Wang [1], Yang Zhang [2]
[1] School of Science, Hebei University of Technology, Tianjin 300401, China
[2] School of Mathematics Science and LPMC, Nankai University, Tianjin, 300071, China.
Corresponding author. zhangyang@nankai.edu.cn
(*Received February 10, 2013, accepted December 23, 2013*)

Abstract. The conservative domain decomposition procedure for solving the variable coefficient diffusion equation is presented. In this procedure, the fluxes at the interface of subdomains are properly defined, which results in the unconditional stability of the procedure. Numerical results examining the stability, the second-order accuracy of solution values as well as fluxes, and parallelism of the procedure are also presented.

Keywords: Diffusion equation; finite difference; domain decomposition; unconditional stability; variable coefficient.

1. Introduction

The diffusion equation is a basic topic, for many equations include the diffusion term, such as Sobolev equation, convection diffusion equations, and so on. The sequent finite difference methods are considered subject to such equations (see [8,9]). There is rich literature on parallel finite difference methods (see [1,2,3,4,5,6,7]). Domain decomposition is a powerful tool for devising parallel methods to solve the diffusion equation. The basic procedure of domain decomposition methods is to first decompose the domain into some subdomains, then define the interface values of subdomains by explicit schemes and the inner values of subdomains by implicit schemes. Once the interface values are available, the global problem is decoupled and parallelization is achieved. Domain decomposition methods with unconditional stability are desired in the application. However, most domain decomposition methods are conditionally stable . The major difficulty devising domain decomposition methods with unconditional stability is defining the suitable interface values of subdomains. It’s also an issue to consider conservative domain decomposition methods, for some diffusion problems have the conservation property. Conservative domain decomposition procedures for the constant coefficient diffusion equation are considered in [1,3,6], in which the scheme in [1] is unconditionally stable. The purpose of this paper is to present the conservative domain decomposition procedure with unconditional stability for the following variable coefficient diffusion problem

$$U_t(x,t)-(\alpha(x,t)U_x(x,t))_x=0, \quad (x,t)\in(0,1)\times(0,T], \tag{1.1}$$

$$U_x(0,t)=U_x(1,t)=0, \quad t\in(0,T], \tag{1.2}$$

$$U(x,0)=U_0(x), \quad x\in[0,1], \tag{1.3}$$

where $\alpha(x,t)$ is smooth enough and $0<\alpha(x,t)\le\bar{\alpha}$. Define the flux

$$Q(x,t)=-\alpha(x,t)U_x(x,t). \tag{1.4}$$

Then (1.1) and (1.2) become as

$$U_t+Q_x=0, \quad (x,t)\in(0,1)\times(0,T], \tag{1.5}$$

and

$$Q(0,t)=Q(1,t)=0, \quad t\in(0,T]. \tag{1.6}$$

From (1.5) and (1.6), there is

$$\frac{d}{dt}\int_0^1 U\,dx=0,$$

which expresses conservation of mass. For the above problem, giving the solution U and flux Q the same second-order accuracy approximations, we consider the block-centered finite difference discretization.

The rest of this paper is organized as follows. In the next section, we present the domain decomposition procedure. In Section 3, we prove the unconditional stability. In Section 4, we examine numerically the stability, accuracy, and parallelism of the procedure. In the final section, we give a conclusion.

2. Domain Decomposition Scheme

Divide the domain $[0,1]\times[0,T]$ by a set of lines parallel to the x - and t -axes. The crossing points are

$$0 = x_{1/2} < x_{3/2} < \cdots < x_{I+1/2} = 1,$$

$$0 = t^0 < t^1 < \cdots < t^N = T.$$

Denote

$$\tau^n = t^n - t^{n-1}, \quad 1 \le n \le N,$$
$$h_i = x_{i+1/2} - x_{i-1/2}, \quad 1 \le i \le I,$$
$$x_i = \frac{x_{i-1/2} + x_{i+1/2}}{2}, \quad 1 \le i \le I,$$
$$h_{i+1/2} = x_{i+1} - x_i = \frac{h_i + h_{i+1}}{2}, \quad 1 \le i \le I-1,$$

and

$$h = \max_i h_i, \quad \hbar = \min_i h_i, \quad \Delta t = \max_n \tau^n.$$

Let f_i^n be the discrete function on $\{(x_i, t^n)\}$ and $f_{i+1/2}^n$ be the discrete function on $\{(x_{i+1/2}, t^n)\}$. Define the difference operators

$$\Delta_\tau f_i^n = \frac{f_i^n - f_i^{n-1}}{\tau^n}, \quad \Delta_+ f_i^n = \frac{f_{i+1}^n - f_i^n}{h_{i+1/2}}, \quad \Delta_- f_{i+1/2}^n = \frac{f_{i+1/2}^n - f_{i-1/2}^n}{h_i},$$

and the discrete norms

$$\| f^n \|^2 = \sum_{i=1}^{I} (f_i^n)^2 h_i, \quad ||| f^n |||^2 = \sum_{i=1}^{I-1} (f_{i+1/2}^n)^2 h_{i+1/2}.$$

Denote $U_i^n = U(x_i, t^n)$, $Q_{i+1/2}^n = Q(x_{i+1/2}, t^n)$. And let u_i^n and $q_{i+1/2}^n$ be the numerical approximations of U_i^n and $Q_{i+1/2}^n$. For simplicity, assume a decomposition of the domain $[0,1]\times[0,T]$ into two subdomains $[0,\overline{x}]\times[0,T]$ and $[\overline{x},1]\times[0,T]$, where $\overline{x} = x_{k+1/2}$ for some integer k, $0<k<I$. It's easy to extend to the case of multiple subdomains. Next we give the domain decomposition procedure.

Approximate the equation (1.5) by

$$\Delta_\tau u_i^n + \Delta_- q_{i+1/2}^n = 0, \quad 1 \le i \le I. \tag{2.1}$$

Enforce the boundary condition (1.6) by

$$q_{1/2}^n = q_{I+1/2}^n = 0, \quad 1 \le n \le N, \tag{2.2}$$

and the initial condition by

$$u_i^0 = U_0(x_i), \quad 1 \le i \le I. \tag{2.3}$$

Sum for (2.1), by the boundary condition (2.2), there is $\sum_{i=1}^{I} u_i^n = \sum_{i=1}^{I} u_i^{n-1}$, which simulates conservation of mass. So, we call the scheme (2.1) conservative scheme.

We further define the approximating values of fluxes. For $1 \le i \le I-1, i \ne k$, approximate $Q_{i+1/2}^n$ by

$$q_{i+1/2}^n = -\alpha_{i+1/2}^n \Delta_+ u_i^n. \tag{2.4}$$

Suppose U and Q are smooth enough, it's easily get from the Taylor expansion that (2.1) and (2.4) have the second-order truncation error $O(h^2 + \Delta t)$ if $\Delta t / \hbar^2$ is any constant. To define $q_{k+1/2}^n$ with the second-order truncation error, from (1.4) and (1.5), we have