INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 3, Number 2, Pages 211–231

A BLOCK MONOTONE DOMAIN DECOMPOSITION ALGORITHM FOR A NONLINEAR SINGULARLY PERTURBED PARABOLIC PROBLEM

IGOR BOGLAEV

(Communicated by Lubin Vulkov)

Abstract. This paper deals with discrete monotone iterative algorithms for solving a nonlinear singularly perturbed parabolic problem. A block monotone domain decomposition algorithm based on a Schwarz alternating method and on a block iterative scheme is constructed. This monotone algorithm solves only linear discrete systems at each time level and converges monotonically to the exact solution of the nonlinear problem. The rate of convergence of the block monotone domain decomposition algorithm is estimated. Numerical experiments are presented.

Key Words. singularly perturbed parabolic problem, block monotone method, nonoverlapping domain decomposition, monotone Schwarz alternating algorithm.

1. Introduction

We are interested in monotone Schwarz alternating algorithms for solving the nonlinear reaction-diffusion problem

$$-\mu^2 (u_{xx} + u_{yy}) + u_t = -f(P,t,u), \tag{1}$$

$$P = (x,y), \quad (P,t) \in Q = \Omega \times (0,T], \quad \Omega = \{0 < x < 1, 0 < y < 1\},$$

$$f_u(P,t,u) \geq 0, \quad (P,t,u) \in \overline{Q} \times (-\infty,\infty), \quad (f_u \equiv \partial f/\partial u),$$

where μ is a small positive parameter. The initial-boundary conditions are defined by

$$u(P,t) = g(P,t),\ (P,t) \in \partial\Omega \times (0,T], \quad u(P,0) = u^0(P),\ P \in \overline{\Omega},$$

where $\partial\Omega$ is the boundary of Ω. The functions $f(P,t,u)$, $g(P,t)$ and $u^0(P)$ are sufficiently smooth. Under suitable continuity and compatibility conditions on the data, a unique solution $u(P,t)$ of (1) exists (see [6] for details). For $\mu \ll 1$, problem (1) is singularly perturbed and characterized by the boundary layers of width $O(\mu|\ln\mu|)$ at the boundary $\partial\Omega$ (see [1] for details).

In the study of numerical solutions of nonlinear singularly perturbed problems by the finite difference method, the corresponding discrete problem is usually formulated as a system of nonlinear algebraic equations. A major point about this system is to obtain reliable and efficient computational algorithms for computing the solution. In the case of the parabolic problem (1), the implicit method is usually in use. On each time level, this method leads to a nonlinear system which requires some kind of iterative scheme for the computation of numerical solutions.

Received by the editors June 20, 2004 and, in revised form, March 24, 2005.

2000 *Mathematics Subject Classification.* 65N06, 65N50, 65N55, 65H10.

A fruitful method for the treatment of these nonlinear systems is the method of upper and lower solutions and its associated monotone iterations (in the case of "unperturbed" problems see [8], [9] and references therein). Since the initial iteration in the monotone iterative method is either an upper or a lower solution, which can be constructed directly from the difference equation without any knowledge of the exact solution (see [2] for details), this method eliminates the search for the initial iteration as is often needed in Newton's method. This elimination gives a practical advantage in the computation of numerical solutions.

In [3], we proposed a discrete iterative algorithm which combines the monotone approach and the iterative domain decomposition method based on the Schwarz alternating procedure. In the case of small values of the perturbation parameter μ, the convergence factor $\tilde{\rho}$ of the monotone domain decomposition algorithm is estimated by

$$\tilde{\rho} = \rho + \mathcal{O}(\tau),$$

where ρ is the convergence factor of the monotone (undecomposed) method and τ is the step size in the t-direction.

The purpose of this paper is to extend the monotone domain decomposition algorithm from [3] in a such way that computation of the discrete linear subsystems on subdomains which are located outside the boundary layers is implemented by the block iterative scheme (see [12] for details of the block iterative scheme). A basic advantage of the block iterative scheme is that the Thomas algorithm can be used for each linear subsystem defined on these subdomains in the same manner as for one-dimensional problems, and the scheme is stable and is suitable for parallel computing.

For solving nonlinear discrete elliptic problems without domain decomposition, the block monotone iterative methods were constructed and studied in [10]. In [10], the convergence analysis does not contain any estimates on a convergence rate of the proposed iterative methods, and the numerical experiments show that these algorithms applied to some model problems converge very slowly. In contrast, a numerical algorithm based on a combination of the domain decomposition approach and the block iterative method applied on subdomains outside the boundary layers converges more quickly than the original block iterative method.

The structure of the paper is as follows. In Section 2, we consider a monotone iterative method for solving the implicit difference scheme which approximates the nonlinear problem (1). In Section 3, we construct and investigate a block monotone domain decomposition algorithm. The rate of convergence of the block monotone domain decomposition algorithm is estimated in Section 4. The final Section 5 presents results of numerical experiments for the proposed algorithm.

2. Monotone iterative method

On $\overline{Q}$ introduce a rectangular mesh $\overline{\Omega}^h \times \overline{\Omega}^\tau$, $\overline{\Omega}^h = \overline{\Omega}^{hx} \times \overline{\Omega}^{hy}$:

$$\overline{\Omega}^{hx} = \left\{x_i,\ 0 \le i \le N_x;\ x_0 = 0,\ x_{N_x} = 1;\ h_{xi} = x_{i+1} - x_i\right\},$$

$$\overline{\Omega}^{hy} = \left\{y_j,\ 0 \le j \le N_y;\ y_0 = 0,\ y_{N_y} = 1;\ h_{yj} = y_{j+1} - y_j\right\},$$

$$\overline{\Omega}^{\tau} = \left\{t_k = k\tau,\ 0 \le k \le N_\tau,\ N_\tau \tau = T\right\}.$$

For a mesh function $U(P,t)$, we use the implicit difference scheme

$$\mathcal{L}^h U(P,t) + \frac{1}{\tau}\left[U(P,t) - U(P,t-\tau)\right] = -f(P,t,U),\ (P,t) \in \Omega^h \times \Omega^\tau, \tag{2}$$

$$U(P,t) = g(P,t),\ (P,t) \in \partial\Omega^h \times \Omega^\tau, \quad U(P,0) = u^0(P),\ P \in \overline{\Omega}^h,$$