Block monotone domain decomposition methods for a quasi-linear anisotropic convection-diffusion equation

I. Boglaev[1] S. Pack[2]

(Received 27 July 2007; revised 14 January 2008)

Abstract

This article deals with discrete monotone iterative methods for solving a quasi-linear, singularly perturbed, convection-diffusion problem. A block monotone domain decomposition method based on a Schwarz alternating method and on block (line) successive under-relaxation iterative method is constructed. The advantages of this monotone method are that the method solves only linear discrete systems at each iterative step of the iterative process and converges monotonically to the exact solution of the quasi-linear problem. Numerical experiments are presented.

See http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/326 for this article, Published April 15, 2008. ISSN 1446-8735

Contents

1 Introduction

We are interested in the semilinear convection-diffusion problem with a boundary layer

$$\begin{aligned} &-\varepsilon u_{xx} - u_{yy} + b(x,y)u_x + f(x,y,u) = 0\,, \quad (x,y) \in \Omega\,, \\ &u = g \text{ on } \partial\Omega\,, \\ &b \geq \beta > 0 \text{ on } \bar{\Omega}\,, \quad 0 < c_* \leq f_u \leq c^*\,, \quad (x,y,u) \in \bar{\Omega} \times (-\infty\,,\infty)\,, \end{aligned} \tag{1}$$

where $\Omega = \{(x,y) : 0 < x < 1, 0 < y < 1\}$, ε is a small positive parameter, β, c_* and c^* are constants, $\partial\Omega$ is the boundary of Ω and $f_u \equiv \partial f/\partial u$.

For $\varepsilon \ll 1$, problem (1) is singularly perturbed and characterized by an elliptic boundary layer of width $\mathcal{O}(\varepsilon|\ln\varepsilon|)$ at $x = 1$. The anisotropic problem (1) frequently occurs in many applications with strongly anisotropic diffusion; for example, this problem is related to the numerical simulation of heat transfer for casting amorphous metal ribbons by melt extraction [2].

This article solves a nonlinear upwind difference scheme applied to (1) by the monotone method (known as the method of lower and upper solutions).

This method leads to iterative methods which converge globally and solve only linear discrete systems at each iterative step which is of great importance in practice.

In the past ten years, with an increase in high performance parallel computers, much interest has been shown in domain decomposition techniques to help reduce the processor time and the computer memory required for solving problems. Domain decomposition techniques involve splitting the problem into subproblems and solving each problem on its own processor. Recently, much interest has been shown in the Schwarz-type iterative domain decomposition methods [4, 5].

The purpose of this article is to extend the monotone domain decomposition method from Boglaev [1] in a such way that computation of the discrete linear subsystems on subdomains which are located outside the boundary layer is implemented by the block (line) successive under-relaxation (BSUR) method (Varga [6] gives details of block iterative methods). A basic advantage of the BSUR method is that the Thomas algorithm can be used for each linear subsystem defined on these subdomains in the same manner as for one-dimensional problem.

In Section 2, for solving (1) we present a nonlinear difference scheme based on an upwind approximation of the first derivative. Section 3 considers the BSUR method which possesses the monotone convergence. Section 4 constructs a block domain decomposition (BDD) method which converges monotonically. The final Section 5 presents results of numerical experiments for the proposed methods.

2 Difference scheme

On $\bar{\Omega} = \Omega \cup \partial\Omega$ we introduce a nonuniform mesh $\bar{\Omega}^h = \bar{\Omega}^{hx} \times \bar{\Omega}^{hy}$:

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

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

For approximation of (1), we use the upwind difference scheme

$$\mathcal{L}^h v(p) + f(p, v) = 0\,, \quad p = (x_i, y_j) \in \Omega^h\,, \quad v = g \text{ on } \partial\Omega^h\,. \tag{2}$$

The linear difference operator

$$\mathcal{L}^h v = -\varepsilon \mathcal{D}_x^2 v - \mathcal{D}_y^2 v + b\mathcal{D}_x^- v\,,$$

where $\mathcal{D}_x^2 v(p)$, $\mathcal{D}_y^2 v(p)$ and $\mathcal{D}_x^- v(p)$ are the central difference and backward difference approximations to the second and first derivatives, respectively,

$$\begin{aligned}
&\mathcal{D}_x^2 v(p) = \left[(v_{i+1,j} - v_{ij})\,/h_{xi} - (v_{ij} - v_{i-1,j})\,/h_{xi-1}\right]/\hbar_{xi}\,,\\
&\mathcal{D}_y^2 v(p) = \left[(v_{i,j+1} - v_{ij})\,/h_{yj} - (v_{ij} - v_{i,j-1})\,/h_{yj-1}\right]/\hbar_{yj}\,,\\
&\mathcal{D}_x^- v(p) = (v_{ij} - v_{i-1,j})\,/h_{xi-1}\,,\\
&\hbar_{xi} = (h_{xi-1} + h_{xi})\,/2\,, \quad \hbar_{yj} = (h_{yj-1} + h_{yj})\,/2\,,
\end{aligned}$$

where $v(p) = v_{ij} = v(x_i, y_j)\,.$

We say that $\bar{v}(p)$ is an upper solution of (2) if it satisfies the inequalities

$$\mathcal{L}^h \bar{v}(p) + f(p, \bar{v}) \ge 0\,, \quad p \in \Omega^h\,, \quad \bar{v} \ge g \quad \text{on } \partial\Omega^h\,.$$

Similarly, $\underline{v}(p)$ is called a lower solution if it satisfies all the reversed inequalities.

3 Monotone BSUR method

We write down the difference scheme (2) at an interior mesh point $(x_i, y_j) \in \Omega^h$ in the form

$$d_{ij} v_{ij} - w_{ij} v_{i-1,j} - e_{ij} v_{i+1,j} - s_{ij} v_{i,j-1} - n_{ij} v_{i,j+1} + f\left(x_i, y_j, v_{ij}\right) + g_{ij}^* = 0\,,$$

where g^*_{ij} is associated with the boundary function $g(p)$, and

$$w_{ij} = \varepsilon/(\hbar_{xi}h_{xi-1}) + b_{ij}/h_{xi-1}\,, \quad e_{ij} = \varepsilon/(\hbar_{xi}h_{xi})\,,$$
$$s_{ij} = 1/(\hbar_{yj}h_{yj-1})\,, \quad n_{ij} = 1/(\hbar_{yj}h_{yj})\,, \quad d_{ij} = w_{ij} + e_{ij} + s_{ij} + n_{ij}\,.$$

The coefficients of the difference scheme satisfy the inequalities

$$d_{ij} > 0\,, \quad w_{ij}\,, e_{ij}\,, s_{ij}\,, n_{ij} \geq 0\,,$$
$$d_{ij} - (w_{ij} + e_{ij} + s_{ij} + n_{ij}) \geq 0\,,$$
$$i = 1, \ldots, N_x - 1\,, \quad j = 1, \ldots, N_y - 1\,.$$

We now define vectors and diagonal matrices by

$$V_i = \left(v_{i,1}, \ldots, v_{i,N_y-1}\right)^T\,, \quad G^*_i = \left(g^*_{i,1}, \ldots, g^*_{i,N_y-1}\right)^T\,,$$
$$F_i\left(V_i\right) = \left(f\left(x_i, y_1, v_{i,1}\right), \ldots, f\left(x_i, y_{N_y-1}, v_{i,N_y-1}\right)\right)^T\,,$$
$$W_i = \operatorname{diag}\left(w_{i,1}, \ldots, w_{i,N_y-1}\right)\,, \quad E_i = \operatorname{diag}\left(e_{i,1}, \ldots, e_{i,N_y-1}\right)\,.$$

Then the difference scheme is represented in the vector-matrix form

$$A_iV_i - W_iV_{i-1} - E_iV_{i+1} + F_i\left(V_i\right) + G^*_i = 0\,, \quad i = 1, \ldots, N_x - 1\,, \qquad (3)$$

with the tridiagonal matrices A_i, $i = 1, \ldots, N_x - 1$,

$$A_i = \begin{bmatrix} d_{i,1} & -n_{i,1} & & & 0 \\ -s_{i,2} & d_{i,2} & -n_{i,2} & & \\ & \ddots & \ddots & \ddots & \\ & -s_{i,N_y-2} & d_{i,N_y-2} & & -n_{i,N_y-2} \\ 0 & & -s_{i,N_y-1} & & d_{i,N_y-1} \end{bmatrix}\,.$$

Matrices W_i and E_i contain the coupling coefficients of a mesh point, respectively, to the mesh point on the left (west) line and the mesh point on the right (east) line.

The block under-relaxation ($0 < \omega \le 1$) or over-relaxation ($1 \le \omega \le 2$) iterative methods for solving the linear system $A_iV_i - W_iV_{i-1} - E_iV_{i+1} + F_i = 0$ have the form [6, p.223]

$$A_iZ_i^{(n)} - \omega W_iZ_{i-1}^{(n)} = -\omega\mathcal{R}_i(V_i^{(n-1)})\,,$$
$$Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}\,, \quad i = 1,\dots,N_x-1\,,$$

where $\mathcal{R}_i(V_i^{(n-1)}) = A_iV_i^{(n-1)} - W_iV_{i-1}^{(n-1)} - E_iV_{i+1}^{(n-1)} + F_i$, and ω is the relaxation parameter. We define the BSUR method for solving the nonlinear system (3) in a similar way with the diagonal preconditioner ωc^*I, where I is the $(N_y-1)\times(N_y-1)$ identity matrix,

$$\begin{aligned}
&(A_i + \omega c^*I)Z_i^{(n)} - \omega W_iZ_{i-1}^{(n)} = -\omega R_i(V_i^{(n-1)})\,, \qquad (4)\\
&Z_i^{(n)} = V_i^{(n)} - V_i^{(n-1)}\,, \quad i = 1,\dots,N_x-1\,, \quad \omega = \text{const} \in (0,1]\,.\\
&R_i(V_i^{(n-1)}) = A_iV_i^{(n-1)} - W_iV_{i-1}^{(n-1)} - E_iV_{i+1}^{(n-1)} + F_i(V_i^{(n-1)}) + G_i^*\,.
\end{aligned}$$

Remark 1 A basic advantage of the BSUR method (4) is that the Thomas algorithm for solving tridiagonal systems can be used for each subsystem i, $i = 1,\dots,N_x-1$. Since $V_0^{(n)}$ is given by the boundary condition, the linear system (4) for $i = 1$ is tridiagonal and is solved for $V_1^{(n)}$ by the Thomas algorithm. Now, the linear system (4) for $i = 2$ is the tridiagonal system for $V_2^{(n)}$. Thus, starting from $i = 1$ and finishing off with $i = N_x - 1$, in serial, we solve only the tridiagonal systems for $V_i^{(n)}$, $i = 1,\dots,N_x-1$.

In the notation $V = (V_1,\dots,V_{N_x-1})^T$, we have the following theorem.

Theorem 2 *Let $\bar{V}^{(0)}, \underline{V}^{(0)}$ be upper and lower solutions of (3). Then the upper sequence $\{\bar{V}^{(n)}\}$ generated by the* BSUR *method (4) converges monotonically from above to the unique solution V of (3), the lower sequence $\{\underline{V}^{(n)}\}$ generated by (4) converges monotonically from below to V:*

$$\underline{V}^{(n-1)} \le \underline{V}^{(n)} \le V \le \bar{V}^{(n)} \le \bar{V}^{(n-1)}\,, \quad n \ge 1\,.$$

Boglaev [3] provides the full proof of this theorem.

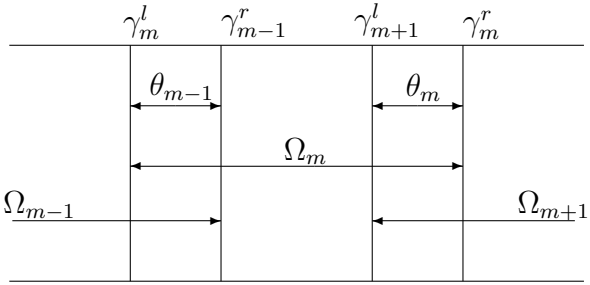

FIGURE 1: Fragment of the domain decomposition with overlapping subdomains Ω_{m-1}, Ω_m, Ω_{m+1} and overlaps θ_{m-1}, θ_m.

4 Monotone BDD method

We now combine the monotone domain decomposition method from Boglaev [1] and the monotone BSUR method (4).

We introduce the set of the overlapping vertical strips $\bar{\Omega}_m$, $m = 1, \ldots, M$, with the boundaries

$$\partial\Omega_m = \gamma^l_m \cup \gamma^r_m \cup \gamma^0_m ,$$

where γ^l_m and γ^r_m are the left and right boundaries of $\bar{\Omega}_m$, respectively, and γ^0_m belongs to the boundary of $\bar{\Omega}$. Thus, $\bar{\Omega}_m \cap \bar{\Omega}_{m+1} = \bar{\theta}_m$, $m = 1, \ldots, M-1$, where $\bar{\theta}_m$ is the overlap between two subdomains $\bar{\Omega}_m$ and $\bar{\Omega}_{m+1}$ (a fragment of the domain decomposition is illustrated in Figure 1). On the subdomains, we introduce nonuniform meshes $\bar{\Omega}^h_m = \bar{\Omega}_m \cap \bar{\Omega}^h$, $m = 1, \ldots, M$.

One complete iterative step includes solving a sequence of M subproblems on subdomains $\bar{\Omega}^h_m$, $m = 1, \ldots, M$, in serial. As the subdomains overlap each other, the left boundary conditions for each subdomain $\bar{\Omega}^h_m$, $m = 2, \ldots, M$, are obtained from the solution found in the previous vertical strip.

1. Initialization: On the whole mesh $\bar{\Omega}^h$, choose an initial function $v^{(0)}(p)$, $p \in \bar{\Omega}^h$, satisfying the boundary condition $v^{(0)}(p) = g(p)$ on $\partial\Omega^h$.

2. On the first subdomain $\bar{\Omega}_1^h$, compute mesh function $z_1^{(n)}(p)$ by using the BSUR method (4), and denote

$$v_1^{(n)}(p) = v^{(n-1)}(p) + z_1^{(n)}(p)\,, \quad p \in \bar{\Omega}_1^h\,. \tag{5}$$

3. On subdomains $\bar{\Omega}_m^h$, $m = 2, \ldots, M$, compute in serial mesh functions $v_m^{(n)}(p)$, $m = 2, \ldots, M$, satisfying the difference schemes

$$\begin{aligned} &(\mathcal{L}^h + c^*) z_m^{(n)}(p) = -\mathcal{R}^h(p, v^{(n-1)})\,, \quad p \in \Omega_m^h\,, \\ &z_m^{(n)}(p) = v_m^{(n)}(p) - v^{(n-1)}(p)\,, \\ &\mathcal{R}^h(p, v^{(n-1)}) = \mathcal{L}^h v^{(n-1)}(p) + f(p, v^{(n-1)})\,, \end{aligned} \tag{6}$$

   with the boundary conditions

$$z_m^{(n)}(\gamma_m^{hl}) = z_{m-1}^{(n)}(\gamma_m^{hl})\,, \quad z_m^{(n)}(\gamma_m^{hr}) = 0\,, \quad z_m^{(n)}(\gamma_m^{h0}) = 0\,,$$

   where

$$\gamma_m^{hl} = \gamma_m^l \cap \bar{\Omega}_m^h\,, \quad \gamma_m^{hr} = \gamma_m^r \cap \bar{\Omega}_m^h\,, \quad \gamma_m^{h0} = \gamma_m^0 \cap \bar{\Omega}_m^h\,.$$

4. Compute the solution $v^{(n)}(p)$, $p \in \bar{\Omega}^h$ by piecing together the solutions on the subdomains

$$v^{(n)}(p) = \begin{cases} v_m^{(n)}(p)\,, & p \in \bar{\Omega}_m^h \setminus \theta_m^h\,,\ m = 1, \ldots, M-1\,; \\ v_M^{(n)}(p)\,, & p \in \bar{\Omega}_M^h\,. \end{cases} \tag{7}$$

5. Stopping criterion: If a prescribed accuracy is reached, then stop; otherwise go to Step 2.

Theorem 3 *Let $\bar{v}^{(0)}, \underline{V}^{(0)}$ be upper and lower solutions of (2). Then the upper sequence $\{\bar{v}^{(n)}\}$ generated by the block domain decomposition (BDD) method (5)–(7) converges monotonically from above to the unique solution v of (2), the lower sequence $\{\underline{V}^{(n)}\}$ generated by method (5)–(7) converges monotonically from below to v:*

$$\underline{V}^{(n-1)}(p) \le \underline{V}^{(n)}(p) \le v(p) \le \bar{v}^{(n)}(p) \le \bar{v}^{(n-1)}(p)\,, \quad p \in \bar{\Omega}^h, \quad n \ge 1\,.$$

Boglaev [3] provides the full proof of this theorem.

Remark 4 In the case where on all subdomains $\bar{\Omega}_m^h$, $m = 1, \ldots, M$, the domain decomposition method (6) and (7) is in use, we get the domain decomposition (DD) method from Boglaev [1]. Theorem 3 holds true for the DD method. Boglaev [1] provides a proof of this result. The particular case $M = 1$ corresponds to the monotone undecomposed method (6).

5 Numerical experiments

Consider the test problem with $b(x, y) = 1$, $f(x, y, u) = 1 - \exp(-u)$ and $g(x, y) = 1$ in (1).

We apply all the three domain decomposition methods, the DD, BSUR and BDD methods, to this problem using a piecewise uniform mesh in the x-direction and uniform mesh in the y-direction. The number of mesh points in the x- and y-direction are set equal to N. This problem has an elliptic boundary layer (area of rapid change) near $x = 1$, where the equation for the width of the boundary layer is given by $\sigma_x = 2\varepsilon \log(N)/\beta$, $\beta = 1$ [1]. We locate half of the mesh points in the x-direction within the layer.

In our numerical experiments, the stopping criteria for the iterates is

$$\max_{P \in \bar{\Omega}^h} \|v^{(n)}(P) - v^{(n-1)}(P)\| \le 10^{-6}\,.$$

To solve the linear difference problems within the monotone DD and the monotone BDD methods, GMRES solver with restarts is used with a diagonal preconditioner. GMRES is a iterative method for finding a numerical solution to a nonsymmetric system in the form $Ax = b$. In the nth iteration, $n = 1, \dots,$ GMRES approximates the solution by using Arnoldi iterations to find a vector in a Krylov subspace K_n with the minimal residual, where $K_n = \mathrm{span}\{b, Ab, A^2b, \dots, A^{n-1}b\}$. One of the disadvantages of a GMRES is the amount of storage required. To overcome this, GMRES with restarts is used. When the solver is restarted all the accumulated data are cleared and the last set of results is used as the initial data. Preconditions are often used within GMRES to accelerate the convergence. Using a preconditioner P, GMRES solves the problem $PAx = Pb$. We use a diagonal preconditioner, where P is the inverse of the diagonal part of A. Within GMRES, the required accuracy used is 10^{-6}, the maximum numbers of iterations and restarts are 50 and 20, respectively.

To decompose the domain $\bar{\Omega}^h$ into M overlapping subdomains we begin by splitting the domain into M subdomains $\bar{Q}^h_m$, $m = 1, \dots, M$,

$$\bar{Q}^h_m = \{(x_i, y_j) : (m-1)N/M \le i \le mN/M\,,\ 0 \le j \le N\}\,,$$
$$\bar{Q}^h_m \cap \bar{Q}^h_{m+1} = \gamma^h_m\,, \quad \gamma^h_m = \{(x_i, y_j) : i = mN/M\,,\ 0 \le j \le N\}\,,$$

where γ^h_m is the interfacial boundary between subdomains $\bar{Q}^h_m$ and $\bar{Q}^h_{m+1}$.

We consider two locations of overlaps. In the first one, we choose overlaps on the left of the interfacial boundaries

$$\bar{\theta}^h_m = \{(x_i, y_j) : mN/M - d \le i \le mN/M\,,\ 0 \le j \le N\}\,, \quad 2 \le d \le N/M - 1\,,$$

where the minimal and maximal overlap sizes are $d = 2$ and $d = N/M - 1$, respectively. Thus, the overlapping subdomains are defined by

$$\bar{\Omega}^h_m = \{(x_i, y_j) : (m-1)N/M - d \le i \le mN/M\,,\ 0 \le j \le N\}\,, \quad m = 2, \dots, M\,,$$

where $\bar{\Omega}^h_1 = \bar{Q}^h_1$. Figure 2(a) illustrates this case. If we choose overlaps on the right of the interfacial boundaries, then in a similar way, we define the

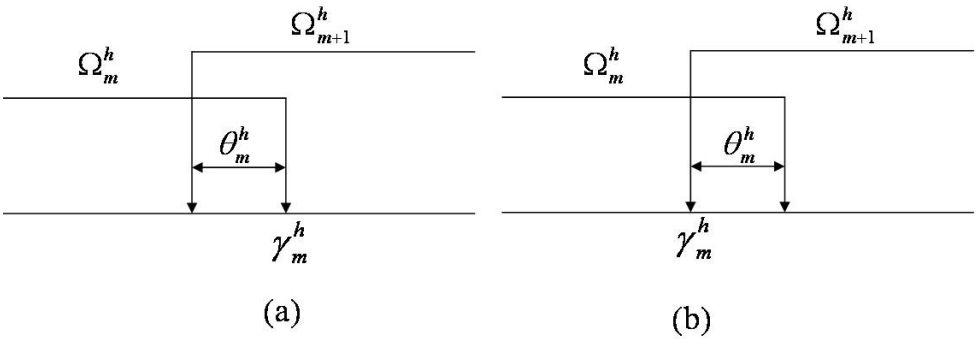

FIGURE 2: (a) Location of the overlap on the left. (b) Location of the overlap on the right.

overlapping subdomains in the form

$$\bar{\Omega}^h_m = \{(x_i, y_j) : (m-1)N/M \le i \le mN/M + d\,,\ 0 \le j \le N\}\,,$$
$$2 \le d \le N/M - 1\,, \quad m = 1, \ldots, M-1\,, \quad \bar{\Omega}^h_M = \bar{Q}^h_M\,.$$

Figure 2(b) illustrates this case.

Table 1 displays the iteration count and execution time of the DD method for the two different locations of the overlaps. Table 1 shows that for the minimal overlap size the iteration count is smaller when the overlaps occur on the left of the interfacial boundaries. However, for the maximal overlap size, there is very little difference in the iteration count of the DD method. For both the minimal and maximal overlap sizes there is little difference in the execution time for the difference locations of the overlaps. For the above reasons the following numerical results have the overlaps located on the left of the interfacial boundaries.

Table 2 displays the iteration counts and execution time over varying numbers of subdomains for different values of ε and N for the monotone DD method. The results are displayed for the minimal and maximal overlap

TABLE 1: Iteration count and execution time of the monotone DD method with the overlaps located to the left and right of the interfacial boundaries using the minimal and maximal overlap sizes above and below the line, respectively.

ε	10^{-1}				10^{-2}				10^{-3}				10^{-4}			
N/M	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
	Iteration count															
Left	6	$\frac{19}{6}$	$\frac{21}{7}$	$\frac{29}{12}$	6	$\frac{7}{6}$	$\frac{21}{7}$	$\frac{32}{13}$	6	$\frac{6}{6}$	$\frac{24}{7}$	$\frac{39}{16}$	6	$\frac{6}{6}$	$\frac{25}{7}$	$\frac{40}{16}$
Right	6	$\frac{19}{6}$	$\frac{21}{7}$	$\frac{29}{12}$	6	$\frac{25}{6}$	$\frac{28}{7}$	$\frac{36}{14}$	6	$\frac{32}{6}$	$\frac{36}{7}$	$\frac{48}{17}$	6	$\frac{33}{6}$	$\frac{37}{7}$	$\frac{49}{18}$
	Execution time (seconds)															
Left	5	$\frac{10}{6}$	$\frac{5}{5}$	$\frac{3}{2}$	10	$\frac{5}{12}$	$\frac{6}{6}$	$\frac{8}{7}$	10	$\frac{4}{12}$	$\frac{8}{6}$	$\frac{9}{7}$	10	$\frac{4}{12}$	$\frac{7}{6}$	$\frac{9}{7}$
Right	5	$\frac{11}{8}$	$\frac{4}{5}$	$\frac{3}{2}$	10	$\frac{10}{12}$	$\frac{7}{6}$	$\frac{8}{7}$	10	$\frac{6}{10}$	$\frac{6}{6}$	$\frac{9}{7}$	10	$\frac{4}{10}$	$\frac{5}{6}$	$\frac{9}{8}$

TABLE 2: Iteration count and execution time of the monotone DD method using the minimal and maximal overlap size above and below the line, respectively.

ε	10^{-1}				10^{-2}				10^{-3}				10^{-4}			
N/M	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
	Iteration count															
2^5	6	$\frac{12}{6}$	$\frac{13}{7}$	$\frac{17}{17}$	6	$\frac{6}{6}$	$\frac{15}{8}$	$\frac{23}{23}$	6	$\frac{6}{6}$	$\frac{17}{8}$	$\frac{27}{27}$	6	$\frac{6}{6}$	$\frac{17}{8}$	$\frac{28}{28}$
2^6	6	$\frac{19}{6}$	$\frac{21}{7}$	$\frac{29}{12}$	6	$\frac{7}{6}$	$\frac{21}{7}$	$\frac{32}{13}$	6	$\frac{6}{6}$	$\frac{24}{7}$	$\frac{39}{16}$	6	$\frac{6}{6}$	$\frac{25}{7}$	$\frac{40}{16}$
2^7	6	$\frac{32}{6}$	$\frac{37}{7}$	$\frac{51}{11}$	6	$\frac{8}{6}$	$\frac{32}{6}$	$\frac{47}{10}$	6	$\frac{13}{6}$	$\frac{37}{7}$	$\frac{58}{12}$	6	$\frac{12}{6}$	$\frac{37}{7}$	$\frac{59}{12}$
	Execution time (seconds)															
2^5	0.4	$\frac{0.4}{0.5}$	$\frac{0.2}{0.3}$	$\frac{0.2}{0.2}$	0.5	$\frac{0.3}{0.7}$	$\frac{0.4}{0.5}$	$\frac{0.5}{0.5}$	0.5	$\frac{0.3}{0.7}$	$\frac{0.5}{0.5}$	$\frac{0.6}{0.6}$	0.5	$\frac{0.3}{0.7}$	$\frac{0.9}{0.9}$	$\frac{0.6}{0.6}$
2^6	5	$\frac{10}{6}$	$\frac{5}{5}$	$\frac{3}{2}$	10	$\frac{5}{12}$	$\frac{6}{6}$	$\frac{8}{7}$	10	$\frac{4}{12}$	$\frac{8}{6}$	$\frac{9}{7}$	10	$\frac{4}{12}$	$\frac{7}{6}$	$\frac{9}{7}$
2^7	140	$\frac{540}{191}$	$\frac{208}{171}$	$\frac{65}{69}$	177	$\frac{150}{254}$	$\frac{163}{179}$	$\frac{91}{89}$	191	$\frac{311}{277}$	$\frac{167}{224}$	$\frac{114}{102}$	191	$\frac{286}{278}$	$\frac{159}{225}$	$\frac{119}{104}$

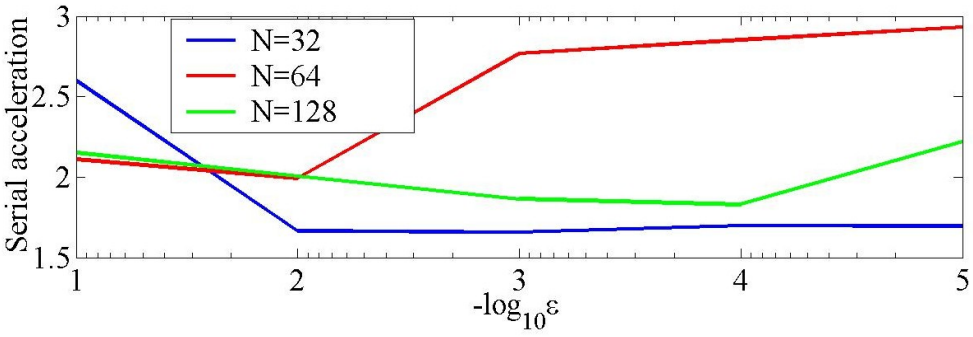

FIGURE 3: Serial acceleration of the monotone DD method.

size above and below the line, respectively. These iteration counts show that for the larger overlap size the iteration count is less. Table 2 also shows that as the number of subdomains increases so does the number of iterations needed for the method to converge. We also conclude from Table 2 that the monotone DD method uniformly converges in its iteration count with respect to ε.

We define the serial acceleration of a domain decomposition method as the execution time of the undecomposed method divided by the minimum execution time of the domain decomposition method. Using the execution times in Table 2, we display the serial acceleration for the monotone DD method in Figure 3.

A serial acceleration greater than 1 indicates an advantage in the domain decomposition method. Figure 3 shows that for all values of ε and N, the serial acceleration is greater than 1 indicating an advantage in using the monotone DD method. As ε decreases the serial acceleration from using the monotone DD method increases.

Figure 4 displays the execution time for the monotone BSUR method over varying values of the relaxation parameter ω, where $N = 64$. Figure 4 shows that $\omega = 1$ (the monotone block Gauss–Seidel method) results in the fastest

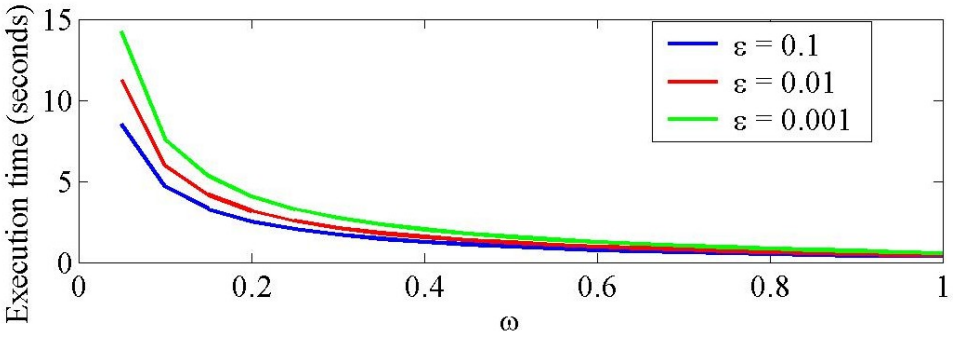

FIGURE 4: Execution time of the monotone BSUR method over varying values of ω.

execution time. Numerical observations indicate that this result holds for varying values of N and ε, leading to the conclusion that $\omega = 1$ is optimal. All following numerical results have $\omega = 1$.

Table 3 displays the iteration count and execution time of the monotone BSUR method for varying values of N and ε. Using Table 3, we see that both the iteration count and the execution time increase with both N and ε^{-1}.

Using Table 3, the serial acceleration is calculated and displayed in Figure 5. Figure 5 shows that for all values of N and ε the serial acceleration is significantly greater than one, indicating a very significant advantage in the monotone BSUR method.

To investigate the serial acceleration of the monotone BDD method, we begin by exploring the effect of varying the size of the first subdomain, which is the subdomain solved by the monotone BSUR method. Figure 6 displays the execution time for varying sizes of the first subdomain with $N = 64$. The top and bottom graphs correspond to two and three subdomains, respectively. In both graphs the minimum overlap size is used. Figure 6 shows that the execution time of the monotone BDD method is minimal when the

TABLE 3: Iteration count and execution time of the monotone BSUR method.

N/ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
	Iteration count				
2^5	86	132	166	171	171
2^6	278	343	449	464	466
2^7	908	923	1229	1279	1284
	Execution time (seconds)				
2^5	0.04	0.05	0.06	0.07	0.07
2^6	0.38	0.46	0.60	0.62	0.62
2^7	4.83	4.88	6.47	6.76	6.78

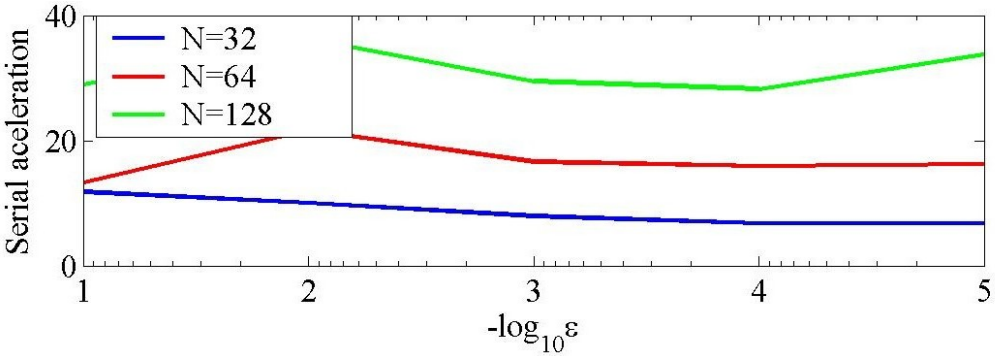

FIGURE 5: Serial acceleration of the monotone BSUR method.

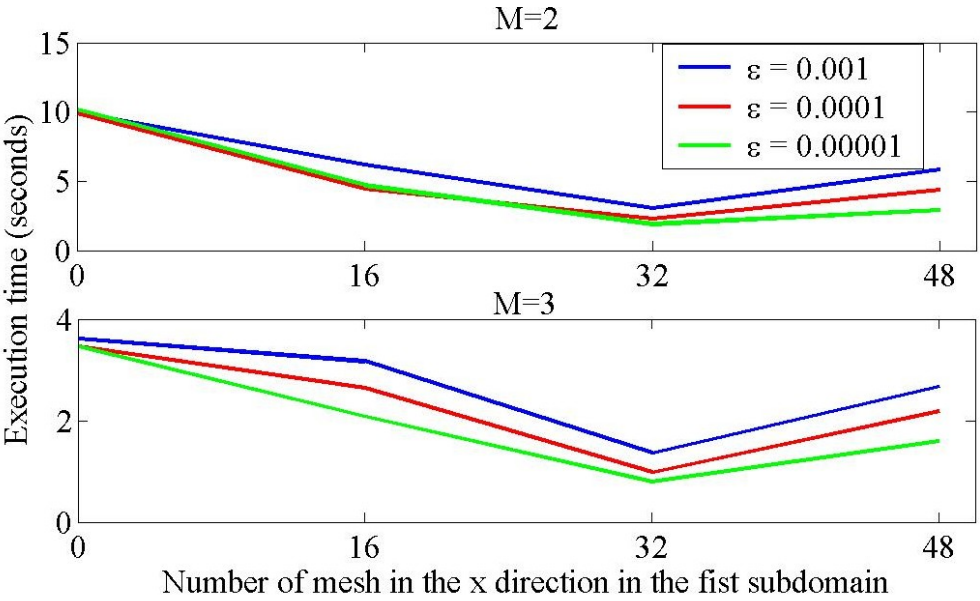

FIGURE 6: Execution time for the monotone BDD method with $N = 64$.

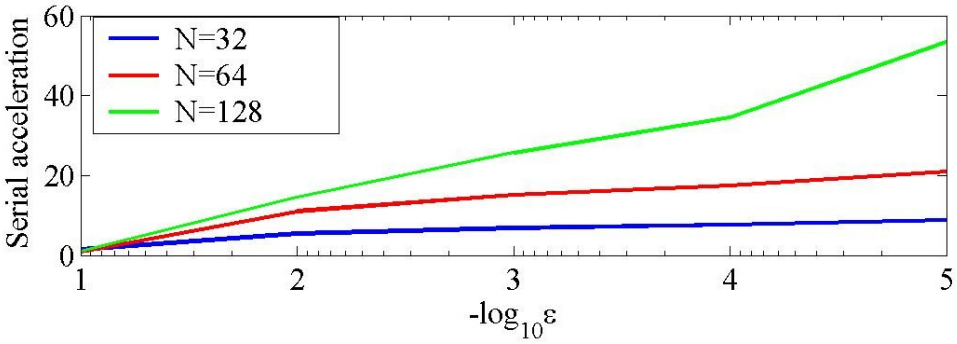

FIGURE 7: Serial acceleration of the monotone BDD method.

first subdomain contains half the amount of mesh points. This is when the monotone BSUR is used outside of the boundary layer and the monotone DD is used within the boundary layer. From our numerical observations this holds for varying values of ε, N and M. Using this information, the monotone BDD method is timed over varying values of ε, N and M with the resulting serial acceleration displayed in Figure 7.

Figure 7 shows that for all values of N and ε there is a significant serial acceleration. It also shows that as ε decreases the serial acceleration for the monotone BDD method increases.

Figure 8 displays the serial acceleration for all the three domain decomposition methods for varying values of ε and different values of N. Figure 8 shows that the monotone DD method results in the smallest serial acceleration. It also shows that when $\varepsilon > 10^{-4}$, the monotone BSUR method results in the highest acceleration. As ε decreases this serial acceleration also tends to decrease and the monotone BDD method results in the fastest serial acceleration.

From the numerical experiments we conclude that all the three domain decomposition methods result in a serial acceleration. The execution time of the monotone BSUR method is minimal when $\omega = 1$. The execution time of

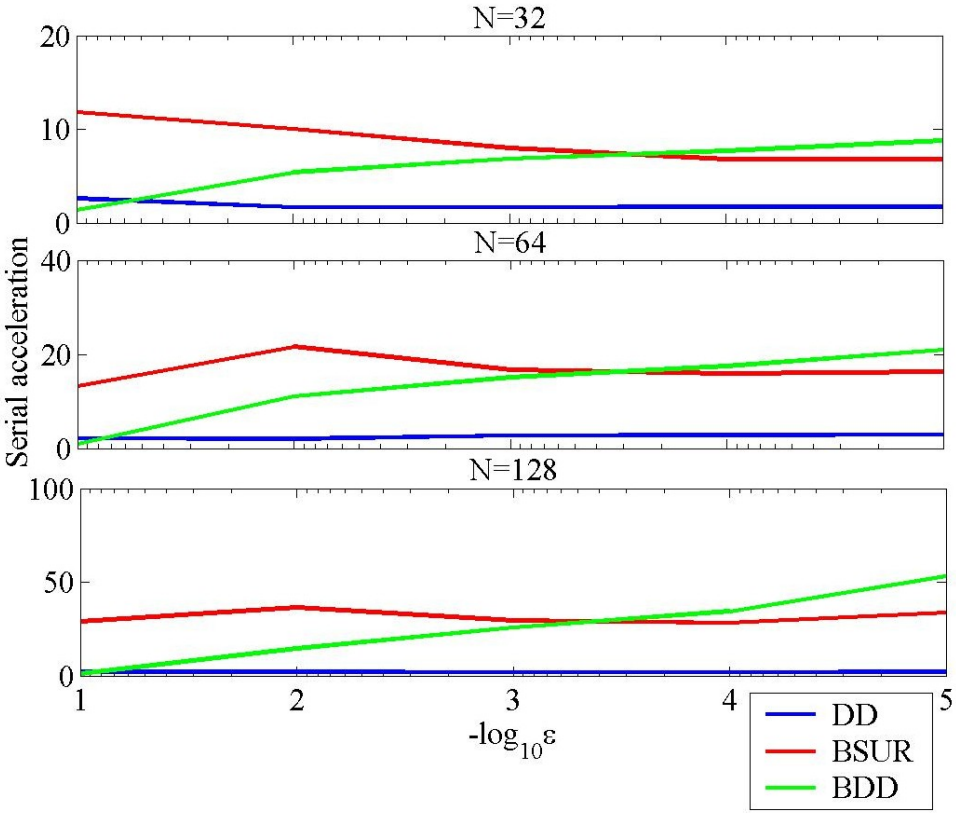

FIGURE 8: Serial acceleration of the monotone domain decomposition methods.

the monotone BDD method is minimal when the first subdomain is outside of the boundary layer. When $\varepsilon > 10^{-4}$, the monotone BSUR method results in the highest serial acceleration. However, when $\varepsilon < 10^{-4}$, the monotone BDD method results in the highest serial acceleration.

References

[1] I. Boglaev, A monotone Schwarz algorithm for a semilinear convection-diffusion problem. *J. Numer. Math.*, 12:169–191, 2004. doi:10.1016/j.cam.2004.03.011 C495, C499, C501

[2] I. P. Boglaev, V. V. Sirotkin, A. Ya. Vilenkin, Ch. V. Kopetskii, A. V. Serebryakov, On the numerical simulation of heat transfer in the process of casting amorphous metal ribbons. Preprint, Institute of Solid State Physics, USSR Academy of Sciences, Chernogolovka, 1983. C494

[3] I. Boglaev and S. Pack, On block monotone domain decomposition algorithms for solving a nonlinear singularly perturbed convection-diffusion problem. Reports in Mathematics 17, IFS, Massey University, 2007. C498, C501

[4] T. Chan and T. Mathew, Domain decomposition algorithms. *Acta Numerica*, 4:61–143, 1994. C495

[5] B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition*. Cambridge University Press, Cambridge, 1996. http://www.cambridge.org/0521602866 C495

[6] R. S. Varga, *Matrix Iterative Analysis*. Second Edition. Springer–Verlag, Berlin Heidelberg, 2000. C495, C498

Author addresses

1. **I. Boglaev**, Institute of Fundamental Sciences, Massey University, Palmerston North, New Zealand.
mailto:i.boglaev@massey.ac.nz

2. **S. Pack**, Institute of Fundamental Sciences, Massey University, Palmerston North, New Zealand.