An iterative domain decomposition algorithm for a nonlinear convection-diffusion problem

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

(Received 3 August 2006; revised 6 September 2007)

Abstract

This article describes an iterative domain decomposition algorithm for solving nonlinear singularly perturbed convection-diffusion problems where convection is dominant. The domain decomposition algorithm consists of the two iterative processes: outer iterations and inner iterations. One outer iterative step represents computing nonlinear difference subproblems on overlapping subdomains in serial according to upwind error propagation (the multiplicative Schwarz method). At the level of the inner iterations, each nonlinear subproblem is solved by the monotone additive Schwarz algorithm. The advantages of the algorithm are that the algorithm solves only linear discrete systems at each iterative step, converges monotonically to the exact solution of the system, and is potentially parallelisable. Results of numerical experiments are presented.

See http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/68 for this article, © Austral. Mathematical Soc. 2007. Published October 18, 2007. ISSN 1446-8735

Contents

1 Introduction

We are interested in iterative domain decomposition methods for solving the semilinear convection-diffusion problem with regular boundary layers:

$$\begin{aligned}&-\varepsilon(u_{xx}+u_{yy})+b(x,y)u_x+f(x,y,u)=0\,,\ \ (x,y)\in\omega\,;\\&u=g \text{ on } \partial\omega\,;\quad b\geq\beta>0 \text{ on } \bar{\omega}\,,\\&f_u\geq c_*>0\,,\quad (x,y,u)\in\bar{\omega}\times(-\infty\,,\infty)\,,\quad (f_u\equiv\partial f/\partial u)\ ,\end{aligned}\qquad(1)$$

where $\bar{\omega}=\omega\cup\partial\omega$, $\omega=\{(x,y):0<x<1 \text{ and } 0<y<1\}$, $\partial\omega$ is the boundary of $\bar{\omega}$, ε is a small positive parameter and β and c_* are constants.

For $\varepsilon\ll 1$, problem (1) is singularly perturbed and characterized by an elliptic boundary layer of width $\mathcal{O}(\varepsilon|\ln\varepsilon|)$ at $x=1$ and by parabolic boundary layers of width $\mathcal{O}(\sqrt{\varepsilon}|\ln\varepsilon|)$ at $y=0$ and $y=1$. The parabolic layers are present because part of the boundary of the domain is a characteristic of the reduced differential equation.

Problem (1) occurs in such fields as fluid dynamics; for example, it describes the magnetohydrodynamic flow in a rectangular duct under a uniform magnetic field at high Hartmann number [7].

In the past ten years, with the increase in high performance parallel computers, many are interested in domain decomposition techniques to help reduce processor time and computer memory required for solving problems. Domain decomposition techniques involve splitting the domain into subproblems and solving each problem on its own processor. Recently, much interest has been shown in the Schwarz-type iterative domain decomposition algorithms [4, 6].

We consider the two level Schwarz domain decomposition algorithm by Garbey et al. [5]. This algorithm consists of the two iterative processes: outer iterations and inner iterations. One outer iterative step represents computing M subproblems on overlapping vertical subdomains (strips) $\bar{\omega}_m$, $m = 1, \ldots, M$, serially, starting from subdomain $\bar{\omega}_1$ and finishing off on $\bar{\omega}_M$ (according to upwind error propagation). Thus, the multiplicative Schwarz method is the outer part of the algorithm. At the level of the inner iterations, each vertical strip $\bar{\omega}_m$ is split into nonoverlapping boxes (horizontal strips) with interface γ. Small interfacial subdomains are introduced near the interface γ, and approximate boundary values computed on γ are used for solving problems on the nonoverlapping box-subdomains. Thus, the additive Schwarz method is the inner part of the algorithm.

The proposed algorithm combines the two level Schwarz domain decomposition algorithm with the method of upper and lower solutions. The method of upper and lower solutions is a monotone iterative method which also provides a method of constructing initial solutions without prior knowledge of the actual solution, as is often required in Newton's method. The monotonicity condition guarantees that systems of algebraic equations based on such methods are well posed. The advantages of the algorithm are that the algorithm solves only linear discrete systems at each iterative step, converges monotonically to the exact solution of the system, and is parallelisable.

2 Nonlinear difference scheme

On $\bar{\omega}$ introduce a nonuniform mesh $\bar{\omega}^h = \bar{\omega}^{hx} \times \bar{\omega}^{hy}$:

$$\begin{aligned} \bar{\omega}^{hx} &= \{x_i\,,\ 0 \le i \le N_x\,;\ x_0 = 0\,,\ x_{N_x} = 1\,;\ h_{xi} = x_{i+1} - x_i\}\,, \\ \bar{\omega}^{hy} &= \{y_j\,,\ 0 \le j \le N_y\,;\ y_0 = 0\,,\ y_{N_y} = 1\,;\ h_{yj} = y_{j+1} - y_j\}\,. \end{aligned}$$

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

$$\mathcal{L}^h U(P) + f(P, U) = 0\,, \quad P = (x_i, y_j) \in \omega^h\,, \quad U = g \text{ on } \partial\omega^h\,, \tag{2}$$

where $\mathcal{L}^h U(P)$ is defined by

$$\mathcal{L}^h U = -\varepsilon \left(\mathcal{D}_x^2 + \mathcal{D}_y^2\right) U + b\mathcal{D}_x^- U\,.$$

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

$$\begin{aligned} \mathcal{D}_x^2 U_{ij} &= \left[\left(U_{i+1,j} - U_{ij}\right)/h_{xi} - \left(U_{ij} - U_{i-1,j}\right)/h_{xi-1}\right]/\hbar_{xi}\,, \\ \mathcal{D}_y^2 U_{ij} &= \left[\left(U_{i,j+1} - U_{ij}\right)/h_{yj} - \left(U_{ij} - U_{i,j-1}\right)/h_{yj-1}\right]/\hbar_{yj}\,, \\ \mathcal{D}_x^- U_{ij} &= \left(U_{ij} - U_{i-1,j}\right)/h_{xi-1}\,, \\ \hbar_{xi} &= \left(h_{xi-1} + h_{xi}\right)/2\,, \quad \hbar_{yj} = \left(h_{yj-1} + h_{yj}\right)/2\,, \end{aligned} \tag{3}$$

where $U_{ij} = U(x_i,\, y_j)$.

3 Domain decomposition algorithm

3.1 The outer iterates

We introduce the set of the overlapping vertical strips $\bar{\omega}_m$, $m = 1, \ldots, M$, with the boundaries (a fragment of the domain decomposition is illustrated on Figure 1)

$$\partial\omega_m = \gamma_m^l \cup \gamma_m^r \cup \gamma_m^0\,,$$

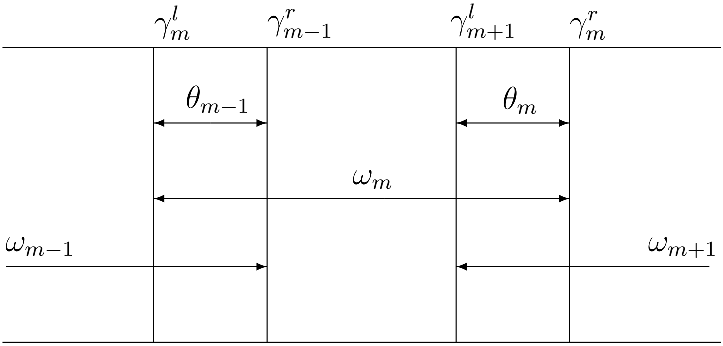

Figure 1: Fragment of the domain decomposition with overlapping subdomains ω_{m-1}, ω_m, ω_{m+1} and overlaps θ_{m-1}, θ_m.

where γ_m^l and γ_m^r are the left and right boundaries of $\bar{\omega}_m$, respectively, and γ_m^0 belongs to the boundary of $\bar{\omega}$. Thus,

$$\bar{\omega}_m \cap \bar{\omega}_{m+1} = \bar{\theta}_m\,, \quad m = 1, \ldots, M-1\,,$$

where $\bar{\theta}_m$ is the overlap between two subdomains $\bar{\omega}_m$ and $\bar{\omega}_{m+1}$. On the subdomains, introduce nonuniform meshes $\bar{\omega}_m^h = \bar{\omega}_m \cap \bar{\omega}^h$, $m = 1, \ldots, M$.

At the level of the outer iterates of the algorithm from Garbey et al. [5], one complete iterative step includes solving a sequence of M problems on subdomains $\bar{\omega}_m^h$, $m = 1, \ldots, M$, in serial.

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 subdomains $\bar{\omega}_m^h$, $m = 1, \ldots, M$, compute in serial the mesh functions $V_m^{(n)}(P)$, $m = 1, \ldots, M$, satisfying the difference schemes

$$\mathcal{L}^h V_m^{(n)}(P) + f(P, V_m^{(n)}) = 0\,, \quad P \in \omega_m^h\,, \tag{4}$$

with the boundary conditions $V_m^{(n)}(\gamma_m^{hl}) = V_{m-1}^{(n)}(\gamma_m^{hl})$, $V_m^{(n)}(\gamma_m^{hr}) = V^{(n-1)}(\gamma_m^{hr})$ and $V_m^{(n)}(\gamma_m^{h0}) = g(\gamma_m^{h0})$, where

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

For computing the problem on subdomain $\bar{\omega}_m^h$, $m > 1$, the Dirichlet boundary condition on the left boundary is updated using the solution of the problem on subdomain $\bar{\omega}_{m-1}^h$ (previous substep of the outer iterative step). Its right boundary is that found from the previous outer iteration $V^{(n-1)}(P)$ and the top and bottom boundaries are equal to the original boundary condition $g(P)$.

3. Compute the solution $V^{(n)}(P)$, $P \in \bar{\omega}^h$, by piecing together the solutions on the subdomains

$$V^{(n)}(P) = \begin{cases} V_m^{(n)}(P)\,, & P \in \bar{\omega}_m^h \setminus \theta_m^h\,, \quad m = 1, \dots, M-1\,; \\ V_M^{(n)}(P)\,, & P \in \bar{\omega}_M^h\,. \end{cases} \tag{5}$$

On $\bar{\omega}_m^h \setminus \theta_m^h$, we set $V^{(n)}$ equal to the solution $V_m^{(n)}$, and overlap $\bar{\theta}_m^h$ is included in $\bar{\omega}_{m+1}^h$.

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

3.2 The inner iterates

We assume that f from (1) satisfies the two-sided constraint

$$0 < c_* \leq f_u \leq c^*\,, \quad c_*, c^* = \text{constant.}$$

For solving the nonlinear problems (4), we use the inner iterates based on the nonoverlapping box-subdomains by Boglaev [2]. We decompose each

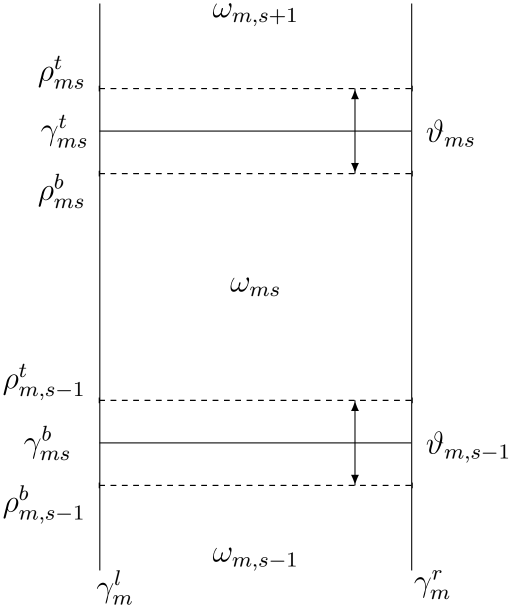

Figure 2: Fragment of the box-domain decomposition.

subdomain $\bar{\omega}_m$ into S_m nonoverlapping box-subdomains ω_{ms}, $s = 1, \ldots, S_m$, with boundaries

$$\partial\omega_{ms} = \gamma^l_{ms} \cup \gamma^r_{ms} \cup \gamma^b_m \cup \gamma^t_{ms} \,,$$

where γ^l_{ms}, γ^r_{ms}, γ^b_m and γ^t_{ms} are the left, right, bottom and top boundaries of $\bar{\omega}_{ms}$. Additionally, we introduce $(S_m - 1)$ interfacial subdomains ϑ_{ms}, $s = 1, \ldots, S_m - 1$, (horizontal strips) with the boundaries

$$\partial\vartheta_{ms} = \rho^l_{ms} \cup \rho^r_{ms} \cup \rho^b_{ms} \cup \rho^t_{ms} \,,$$

where ρ^l_{ms}, ρ^r_{ms}, ρ^b_{ms} and ρ^t_{ms} are the left, right, bottom and top boundaries of $\bar{\vartheta}_{ms}$. Figure 2 illustrates a fragment of the box-domain decomposition.

On each iterative step of the inner iterates, we first solve problems on the nonoverlapping subdomains $\bar{\omega}^h_{ms} = \bar{\omega}_{ms} \cap \bar{\omega}^h_m$, $s = 1, \ldots, S_m$, with Dirichlet boundary conditions passed from the previous iterate. Then Dirichlet data are passed from these subdomains to the horizontal interfacial subdomains

$\bar{\vartheta}^h_{ms} = \bar{\vartheta}_{ms} \cap \bar{\omega}^h_m$, $s = 1, \ldots, S_m - 1$. Problems on the horizontal interfacial subdomains are computed. Finally, we piece together the solutions on the subdomains.

1. Initialization: On $\bar{\omega}^h_m$, choose an initial mesh function $V^{(n,0)}_m(P)$, $P \in \bar{\omega}^h_m$, satisfying the boundary conditions in (4).
2. On the box-subdomains $\bar{\omega}^h_{ms}$, $s = 1, \ldots, S_m$, compute the mesh functions $Z^{(n,k)}_{ms}(P)$ (here the index k stands for a number of the inner iterative steps) satisfying the difference problems
$$(\mathcal{L}^h + c^*)Z^{(n,k)}_{ms}(P) = -\mathcal{R}^h(P, V^{(n,k-1)}_m) , \quad P \in \omega^h_{ms} , \tag{6}$$
with $Z^{(n,k)}_{ms}(\partial\omega^h_{ms}) = 0$, where $\mathcal{R}^h(P, V^{(n,k-1)}_m)$ is the residual of the difference scheme (4) on $V^{(n,k-1)}_m$, that is,
$$\mathcal{R}^h(P, V^{(n,k-1)}_m) = \mathcal{L}^h V^{(n,k-1)}_m(P) + f(P, V^{(n,k-1)}_m) .$$
3. On the horizontal interfacial subdomains $\bar{\vartheta}^h_{ms}$, $s = 1, \ldots, S_m - 1$, compute the difference problems
$$(\mathcal{L}^h + c^*)\widetilde{Z}^{(n,k)}_{ms}(P) = -\mathcal{R}^h(P, V^{(n,k-1)}_m) , \quad P \in \vartheta^h_{ms} , \tag{7}$$
with $\widetilde{Z}^{(n,k)}_{ms}(\rho^{hl}_{ms}) = \widetilde{Z}^{(n,k)}_{ms}(\rho^{hr}_{ms}) = 0$, $\widetilde{Z}^{(n,k)}_{ms}(\rho^{hb}_{ms}) = Z^{(n,k)}_{ms}(\rho^{hb}_{ms})$ and $\widetilde{Z}^{(n,k)}_{ms}(\rho^{ht}_{ms}) = Z^{(n,k)}_{m,s+1}(\rho^{ht}_{ms})$.
4. Compute the mesh function $V^{(n,k)}_m(P)$, $P \in \bar{\omega}^h_m$ by piecing together the solutions on the subdomains
$$V^{(n,k)}_m(P) = \begin{cases} V^{(n,k-1)}_m(P) + Z^{(n,k)}_{ms}(P) , & P \in \bar{\omega}^h_{ms} \setminus \left(\bar{\vartheta}^h_{m,s-1} \cup \bar{\vartheta}^h_{ms}\right) ; \\ V^{(n,k-1)}_m(P) + \widetilde{Z}^{(n,k)}_{ms}(P) , & P \in \bar{\vartheta}^h_{ms}, \ s = 1, \ldots, S_m - 1 . \end{cases} \tag{8}$$
5. Stopping criterion: If a prescribed accuracy is reached, then stop; otherwise go to Step 2.

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

$$\mathcal{L}^h\bar{U}(P)+f(P,\bar{U})\geq 0\,,\quad P\in\omega^h\,,\quad \bar{U}\geq g \text{ on } \partial\omega^h\,.$$

Similarly, $\underline{U}(P)$ is called a lower solution if it satisfies the reversed inequalities. If the initial mesh function $V_m^{(n,0)}$ is an upper or lower solution on subdomain $\bar{\omega}_m^h$, then the sequence generated by (6)–(8) converges monotonically from above or below, respectively, to $V_m^{(n)}$. This is proven in a similar way as Theorem 3 by Boglaev [2].

Algorithm (6)–(8) can be carried out by parallel processing. Each of the box-subdomain problems $\bar{\omega}_{ms}$, $s=1,\ldots,S_m$, in Step 2 can be solved on their own processor in parallel. Next in Step 3, each of the interfacial subdomain problems $\bar{\vartheta}_{ms}^h$, $s=1,\ldots,S_m-1$, can be solved on their own processors in parallel. Being able to construct algorithms for use on parallel computers aids in the reduction of problems caused by processor time and computer memory.

3.3 Numerical stability of the outer and inner iterates

Here we show that the combined domain decomposition algorithm, based on the outer (4), (5) and inner iterates (6)–(8), possesses numerical stability.

On each subdomain $\bar{\omega}_m^h$, $m=1,\ldots,M$, let the stopping criterion for the inner iterates (6)–(8) be defined in the form

$$\|\mathcal{R}^h(P,V_m^{(n,k)})\|_{\omega_m^h}\leq\Delta\,, \tag{9}$$

where $\|\cdot\|_{\omega_m^h}$ is the maximal norm and Δ is a prescribed accuracy.

The solution generated by the combined domain decomposition algorithm (4), (5), (6)–(8) with the stopping criterion (9) is denoted by $\widehat{V}^{(n)}(P)$, $P\in\bar{\omega}^h$.

Theorem 1 *The combined domain decomposition algorithm (4), (5), (6)–(8), with the stopping criterion (9), is numerically stable:*

$$\|\widehat{V}^{(n)} - V^{(n)}\|_{\bar{\omega}^h} \leq \Delta/c^* , \quad n \geq 1 , \tag{10}$$

where $\|\cdot\|_{\bar{\omega}^h}$ is the maximal norm and $V^{(n)}(P)$ is the solution generated by the outer iterates (4) and (5).

Boglaev and Pack [3] gave the full proof of the theorem.

4 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 the domain decomposition algorithm (4), (5), (6)–(8), to this problem using a piecewise uniform mesh in the x-direction and a uniform mesh in the y-direction. As this problem has an elliptic boundary layer near $x = 1$, half the mesh points are used within this layer. The equation for the width of the boundary layer is $\sigma_x = 2\varepsilon\log(N_x)/\beta$, $\beta = 1$ [1].

For the following numerical results, the stopping criterion for the outer iterates is defined by

$$\left|\left|V^{(n)} - V^{(n-1)}\right|\right|_{\bar{\omega}^h} \leq \Delta ,$$

and the stopping criterion for the inner iterates is defined in (9), where Δ is the required accuracy. For this test problem the required accuracy is $\Delta = 10^{-6}$.

For the following experiments, we focus on balanced domain decompositions where M is even and $M/2$ vertical strips are placed within the boundary layer. Balanced domain decompositions are more suited to parallel implementation than unbalanced domain decompositions. The overlap between

Table 1: Outer iteration count using the minimum and maximum overlap size, above and below the line, respectively.

ε	10^{-2}			10^{-3}			10^{-4}		
$N\backslash M$	2	4	8	2	4	8	2	4	8
	Convergence Iteration Count								
2^5	$\frac{6}{3}$	$\frac{10}{5}$	$\frac{13}{13}$	$\frac{4}{3}$	$\frac{10}{5}$	$\frac{13}{13}$	$\frac{3}{3}$	$\frac{10}{5}$	$\frac{13}{13}$
2^6	$\frac{7}{3}$	$\frac{13}{4}$	$\frac{17}{7}$	$\frac{4}{3}$	$\frac{13}{4}$	$\frac{17}{7}$	$\frac{3}{3}$	$\frac{13}{4}$	$\frac{17}{7}$
2^7	$\frac{10}{3}$	$\frac{19}{4}$	$\frac{25}{5}$	$\frac{4}{3}$	$\frac{19}{4}$	$\frac{25}{5}$	$\frac{3}{3}$	$\frac{19}{4}$	$\frac{25}{6}$

the vertical strips is chosen so that for the two vertical strips either side of the boundary layer, the overlap occurs outside of the boundary layer.

The number of mesh points in the x-direction and y-direction are set equal to N, and the width of the interfacial subdomains is held fixed at $N/(2S)$, where $S = S_m$, $m = 1, \ldots, M$.

For the tables presented, results for the minimal and maximal size of the overlaps appear above and below the line, respectively.

Table 1 shows the outer iteration counts over varying numbers of vertical strips and for different values of ε and N. In our numerical experiments the number of horizontal strips did not affect the outer iteration count.

The outer iteration counts in Table 1 show that for the larger overlap size the outer iteration count is less. As the number of vertical strips increases, so does the number of outer iterations needed for the algorithm to converge. Table 1 also shows that the domain decomposition algorithm uniformly converges in its outer iteration count with respect to ε.

Table 2 displays the serial execution time over varying numbers of vertical and horizontal strips, for different values of ε and N. From these results we observe that the execution time is smaller for the maximal overlap size

Table 2: Execution time using the minimum and maximum overlap size, above and below the line, respectively.

	ε	10^{-2}				10^{-3}				10^{-4}			
N	$S\backslash M$	1	2	4	8	1	2	4	8	1	2	4	8
		Execution time (seconds)											
2^5	1	0.4	$\frac{0.4}{0.2}$	$\frac{0.4}{0.2}$	$\frac{0.4}{0.4}$	0.3	$\frac{0.2}{0.2}$	$\frac{0.3}{0.1}$	$\frac{0.3}{0.3}$	0.3	$\frac{0.2}{0.1}$	$\frac{0.2}{0.1}$	$\frac{0.2}{0.2}$
	2	0.4	$\frac{0.6}{0.2}$	$\frac{0.6}{0.3}$	$\frac{0.6}{0.6}$	0.4	$\frac{0.3}{0.2}$	$\frac{0.4}{0.2}$	$\frac{0.4}{0.4}$	0.3	$\frac{0.2}{0.2}$	$\frac{0.3}{0.1}$	$\frac{0.3}{0.3}$
	4	0.4	$\frac{0.6}{0.3}$	$\frac{0.7}{0.4}$	$\frac{0.9}{0.9}$	0.4	$\frac{0.4}{0.2}$	$\frac{0.5}{0.2}$	$\frac{0.6}{0.6}$	0.4	$\frac{0.3}{0.2}$	$\frac{0.5}{0.2}$	$\frac{0.5}{0.5}$
	8	0.4	$\frac{0.7}{0.4}$	$\frac{1.1}{0.7}$	$\frac{1.9}{1.9}$	0.3	$\frac{0.4}{0.2}$	$\frac{0.7}{0.3}$	$\frac{1.0}{1.0}$	0.3	$\frac{0.3}{0.2}$	$\frac{0.6}{0.2}$	$\frac{0.9}{0.9}$
2^6	1	13.8	$\frac{6.2}{3.4}$	$\frac{4.4}{1.9}$	$\frac{3.5}{2.2}$	11.2	$\frac{3.2}{3.3}$	$\frac{2.7}{1.1}$	$\frac{2.2}{1.0}$	16.5	$\frac{2.9}{2.7}$	$\frac{2.6}{0.8}$	$\frac{1.9}{0.7}$
	2	8.5	$\frac{7.9}{3.2}$	$\frac{6.1}{2.5}$	$\frac{5.4}{3.1}$	9.9	$\frac{4.2}{2.7}$	$\frac{4.1}{1.5}$	$\frac{3.3}{1.5}$	12.8	$\frac{3.5}{2.5}$	$\frac{3.4}{1.0}$	$\frac{2.8}{0.9}$
	4	5.9	$\frac{7.5}{3.1}$	$\frac{6.3}{2.5}$	$\frac{6.4}{3.5}$	10.2	$\frac{4.5}{2.8}$	$\frac{5.0}{1.7}$	$\frac{4.6}{1.9}$	13.1	$\frac{3.7}{2.7}$	$\frac{4.2}{1.2}$	$\frac{4.0}{1.3}$
	8	4.5	$\frac{7.7}{3.9}$	$\frac{8.1}{3.5}$	$\frac{10.6}{5.5}$	8.1	$\frac{4.6}{2.9}$	$\frac{5.7}{1.9}$	$\frac{6.4}{2.6}$	12.7	$\frac{4.0}{2.7}$	$\frac{5.1}{1.4}$	$\frac{6.1}{1.9}$
2^7	1	226.4	$\frac{619.4}{134.0}$	$\frac{114.6}{52.4}$	$\frac{49.8}{24.5}$	289.2	$\frac{239.8}{150.0}$	$\frac{56.2}{36.8}$	$\frac{26.9}{11.4}$	627.2	$\frac{372.9}{163.7}$	$\frac{61.1}{33.4}$	$\frac{25.8}{8.4}$
	2	203.1	$\frac{415.1}{105.3}$	$\frac{94.2}{37.8}$	$\frac{68.7}{24.1}$	326.9	$\frac{176.0}{135.9}$	$\frac{49.6}{27.9}$	$\frac{39.5}{12.4}$	662.8	$\frac{245.8}{160.8}$	$\frac{48.1}{24.4}$	$\frac{37.1}{8.8}$
	4	106.8	$\frac{250.1}{66.1}$	$\frac{95.6}{32.1}$	$\frac{73.9}{24.9}$	202.9	$\frac{131.7}{97.1}$	$\frac{53.5}{23.8}$	$\frac{49.7}{14.2}$	387.9	$\frac{170.0}{107.0}$	$\frac{51.3}{20.9}$	$\frac{47.2}{10.3}$
	8	63.8	$\frac{186.7}{62.3}$	$\frac{106.6}{40.7}$	$\frac{96.2}{31.0}$	140.0	$\frac{132.5}{83.6}$	$\frac{58.5}{24.0}$	$\frac{63.8}{16.3}$	265.8	$\frac{164.7}{88.2}$	$\frac{56.0}{21.4}$	$\frac{62.1}{12.4}$

compared to that of the minimal overlap size.

The highlighted results in Table 2 are the minimal execution times for the maximal and minimal overlap size in each column for $N = 128$. These show that the absolute minimum execution time, for both the minimal and maximal overlap size, for all ε, occurs when the number of vertical strips is 8 and the number of horizontal strips is 1. By using these execution times, the acceleration (minimum execution time of the domain decomposition algorithm/ execution time of the undecomposed algorithm) of the domain decomposition algorithm is 9, 25, 75, for $\varepsilon = 10^{-2}$, 10^{-3}, 10^{-4}, respectively.

Table 2 shows that for a fixed number of vertical strips the ideal number of horizontal strips is the same for all values of ε. For example, for $M = 2$, 4, 8, the minimum execution time occurs for both the minimal and maximal overlap size when $S = 8$, 4, 1, respectively.

From our numerical experiments we draw the following conclusions.

- The outer iteration count is uniformly convergent in ε.
- The execution time of the domain decomposition algorithm decreases as the overlap size increases.
- The serial execution times for the domain decomposition algorithm show a considerable acceleration compared to the undecomposed method.

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 C503

[2] I. Boglaev, On monotone iterative methods for a nonlinear singularly perturbed reaction-diffusion problem. *J. Comput. Appl. Math.*, 162:445–466, 2004. doi:10.1016/j.cam.2003.08.035 C499, C502

[3] I. Boglaev and S. Pack, An iterative domain decomposition algorithm for a nonlinear convection-diffusion problem. Reports in Mathematics **16**, IFS, Massey University, 2006. C503

[4] T. Chan and T. Mathew, Domain decomposition algorithms. *Acta Numerica*, 4:61–143, 1994. http://citeseer.ist.psu.edu/chan94domain.html C496

[5] M. Garbey, Yu. A. Kuznetsov, and Yu. V. Vassilevski, A parallel Schwarz method for a convection-diffusion problem. *SIAM J. Sci. Comput.*, 22:891–916, 2000. doi:10.1137/S1064827598335854 C496, C498

[6] B. Smith, P. Bjørstad, and W. Gropp, *Domain decomposition.* Cambridge University Press, Cambridge, 1996. http://www.cambridge.org/0521602866 C496

[7] D. J. Temperley and L. Todd, The effects of wall conductivity in magnetohydrodynamic duct flow at high Hartmann numbers, *Proc. Cambridge Philos. Soc.*, 69:337–351, 1971. C496

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.
   mailto:s.pack@massey.ac.nz