Solution of Singular Perturbation Problems via the Domain Decomposition Method on Serial and Parallel Computers

Paul A. Farrell, Igor P. Boglaev and Vadim V. Sirotkin

Computer Science Program
Technical Report Number CS1994-01-12
August 31, 1995

Department of Mathematics and Computer Science
Kent State University
Kent, Ohio 44242.

Parallel Domain Decomposition Methods for Semi-linear Singularly Perturbed Differential Equations

Paul A. Farrell*
Department of Mathematics and Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Igor P. Boglaev Vadim V. Sirotkin
Institute of Microelectronics Technology
Russian Academy of Sciences, Moscow District, 142432
Chernogolovka, Russia.

August 31, 1995

Abstract:- Domain decomposition methods for the solution of semi-linear selfadjoint and non-selfadjoint singularly perturbed differential equations are considered. An iterative domain decomposition algorithm suitable for parallelization is described and convergence of the algorithm is established. The implementation on a shared memory multiprocessor is described and numerical examples are presented to demonstrate the effectiveness of the algorithm.

1. Introduction

We consider iterative domain decomposition algorithms for the solution of semi-linear selfadjoint and non-selfadjoint singularly perturbed two-point boundary value problems. These reduce the given problem to sequences of boundary value problems on appropriate subdomains. Iterative domain decomposition algorithms, which are suitable for parallelization have been described previously in [3, 4]. In this paper, we describe iterative algorithms, which are suitable for implementation on shared memory or message passing MIMD computers. We prove that these converge to the solution of the original problem and derive the rate of convergence. We illustrate the methods by implementing the algorithm using fitted finite difference methods on special non-uniform grids. The methods are implemented on a 26 processor Sequent Balance shared memory MIMD computer, and results are presented

*Supported by The Research Council of Kent State University

giving the speed-up of the algorithm compared to the usual sequential domain decomposition algorithm.

We consider the following two semi-linear singular perturbation problems. The first is the selfadjoint problem

$$L_\epsilon u(x) \equiv \epsilon u'' = f(x,u), \qquad x \in \Omega, \qquad \Omega = (0,1), \tag{1a}$$

$$u(0) = u_0, \qquad u(1) = u_1, \tag{1b}$$

$$f_u \geq \beta_0^2, \ \beta_0 = \text{const} > 0, \qquad (f_u = \partial f / \partial u), \tag{1c}$$

where $\epsilon \equiv \mu^2$, μ is a small positive parameter. The solution of (1a)-(1c) has boundary layers at $x = 0$ and $x = 1$. The width of these boundary layers is of the order of $h_\epsilon = \mu \mid \ln(\mu) \mid / \beta_0$. For simplicity, we assume in this paper that the solution $u(x)$ exhibits a boundary layer only at $x = 0$ (that is that the reduced solution satisfies the boundary condition (1b)).

The second problem is the non-selfadjoint problem

$$L_\epsilon u(x) \equiv \epsilon u'' + \alpha(x) u' = f(x,u), \qquad x \in \Omega, \qquad \Omega = (0,1), \tag{2a}$$

$$u(0) = u_0, \qquad u(1) = u_1, \tag{2b}$$

$$\alpha(x) \geq \alpha_0 = \text{const} > 0, \qquad f_u \geq 0, \tag{2c}$$

where ϵ is a small positive parameter. The solution of (2a)-(2c) has a boundary layer at $x = 0$ of width $h_\epsilon = \epsilon \mid \ln(\epsilon) \mid / \alpha_0$.

In this paper we consider two iterative algorithms based on domain decomposition for the solution of the semi-linear singularly perturbed problems (1) and (2). Domain decompositions and iterative algorithms for these problems are introduced in section 2. Section 3 gives the proof of the convergence results for these algorithms. In section 4 we present numerical examples and compare the performance of the serial and parallel versions of these iterative algorithms.

2. Iterative Algorithms

Consider the decomposition of the domain $\Omega = (0,1)$ into two overlapping subdomains Ω_1 and Ω_2:

$$\Omega_1 = (0, x_r), \qquad \Omega_2 = (x_l, 1), \qquad 0 < x_l < x_r < 1.$$

We now introduce two sequences of functions $\{v^n(x)\}$, $\{w^n(x)\}$, $n \geq 1$, satisfying the equations:

$$L_\epsilon v^n(x) = f(x, v^n), \qquad x \in \Omega_1, \quad v^n(0) = u_0, \quad v^n(x_r) = v_r^n, \tag{3a}$$

$$L_\epsilon w^n(x) = f(x, w^n), \qquad x \in \Omega_2, \quad w^n(x_l) = w_l^n, \quad w^n(1) = u_1. \tag{3b}$$

Here L_ϵ is defined by (1a) or (2a) and $u_0,\ u_1$ by (1b) or (2b), respectively.

We now construct the two iterative algorithms. The first, A1, is the normal Schwarz alternating procedure. The boundary conditions $v_r^n,\ w_l^n$ from (3a), (3b) are defined by

$$v_r^{n+1} = w^n(x_r), \qquad w_l^n = v^n(x_l), \tag{4}$$

where an initial guess v_r^1 must be prescribed.

The second algorithm, A2, is constructed using the interfacial problem

$$L_\epsilon z^n(x) = f(x, z^n), \qquad x \in \Omega_{\text{inf}} = (X_l, X_r), \tag{5a}$$

$$z^n(X_l) = v^n(X_l), \qquad z^n(X_r) = w^n(X_r),$$

where $X_l < x_l < x_r < X_r$. Here the boundary conditions from $(3a), (3b)$ are determined by the following condition rather than by (4),

$$v_r^{n+1} = z^n(x_r), \qquad w_l^{n+1} = z^n(x_l), \tag{5b}$$

where the initial guesses v_l^1 and w_r^1 are given.

Algorithm A1 is a serial procedure, since the solution v^n of (3a) must be obtained in order to determine the boundary condition $w_l^n = v^n(x_l)$ used in (3b). Thus (3a) and (3b) are executed in lockstep fashion. Algorithm A2 can however be carried out by parallel processing, since on each iteration step problems (3a) and (3b) can be solved concurrently to give both v^n and w^n. The solution of the interfacial problem (5a), (5b) represents the sequential part of the algorithm.

3. Convergence of the Iterative Algorithms

In this section we formulate and prove convergence results for algorithms A1 and A2. In the following lemmas, we obtain some technical results required later.

We introduce two linear two-point boundary value problems related to (1) and (2) given by :

$$L_\epsilon y(x) - b(x)y(x) = 0, \quad x \in \Omega = (x_1, x_2), \tag{6a}$$

$$y(x_1) = y_1, \quad y(x_2) = y_2, \tag{6b}$$

where the coefficient $b(x)$ satisfies the conditions

$$b(x) \geq b_0 = \begin{cases} \beta_0^2, & \text{if } L_\epsilon \text{ from (1)}, \\ 0, & \text{if } L_\epsilon \text{ from (2)}. \end{cases} \tag{6c}$$

Denote by $\varphi_\Omega^{I,II}(x)$ the solutions of the linear problems

$$L_\epsilon \varphi_\Omega^I - b_0 \varphi_\Omega^I = 0,\ x \in \Omega, \tag{7a}$$

$$\varphi_\Omega^I(x_1) = 1,\ \varphi_\Omega^I(x_2) = 0.$$

and

$$L_\epsilon \varphi_\Omega^{II} - b_0 \varphi_\Omega^{II} = 0,\ x \in \Omega, \tag{7b}$$
$$\varphi_\Omega^{II}(x_1) = 0,\ \varphi_\Omega^{II}(x_2) = 1.$$

respectively.

Lemma 1. *If $y(x)$ is the solution to (6), then for all $x \in \bar{\Omega}$ we have the estimate*

$$\mid y(x) \mid \leq \varphi_\Omega^I(x) \mid y_1 \mid + \varphi_\Omega^{II}(x) \mid y_2 \mid, \tag{8}$$

where $\varphi_\Omega^{I,II}$ are given by (7a) and (7b) respectively.

Proof: Let $Y(x)$ be the solution of the linear problem

$$L_\epsilon Y(x) - b_0 Y(x) = 0, \quad x \in \Omega, \quad Y(x_1) = \mid y_1 \mid, \quad Y(x_2) = \mid y_2 \mid . \tag{9}$$

From the maximum principle for the operator $(L_\epsilon - b_0)$ we conclude that $Y(x) \geq 0, x \in \Omega$. From (6) and (9) we have

$$L_\epsilon(Y \pm y) - b(Y \pm y) = (b_0 - b)Y, \quad x \in \Omega, \quad (Y \pm y)_{x_1, x_2} \geq 0.$$

Using (6c) and the inequality $Y \geq 0$, by the maximum principle for the operator $(L_\epsilon - b)$, it follows that $Y \pm y \geq 0, x \in \bar{\Omega}$. This is equivalent to $\mid y(x) \mid \leq Y(x),\ x \in \bar{\Omega}$. Since the solution of problem (9) can be written in the form

$$Y(x) = \varphi_\Omega^I(x) \mid y_1 \mid + \varphi_\Omega^{II}(x) \mid y_2 \mid, \quad x \in \bar{\Omega},$$

we obtain the required estimate (8). □

Lemma 2. *The solutions $\varphi_\Omega^{I,II}(x)$ of problems (7a) and (7b) satisfy the following inequalities*

$$0 < \varphi_\Omega^{I,II}(x) < 1, x \in \Omega; \tag{10a}$$

$$c^I \varphi_\Omega^I(x) + c^{II} \varphi_\Omega^{II}(x) \leq \max(c^I, c^{II}), \quad x \in \bar{\Omega}, \tag{10b}$$

where coefficients $c^I,\ c^{II} \geq 0$;

$$\varphi_\Omega^I(x) \leq \begin{cases} \exp[-\beta_0(x - x_1)/\mu], & \text{if } L_\epsilon \text{ from (1)}, \\ \exp[-\alpha_0(x - x_1)/\epsilon], & \text{if } L_\epsilon \text{ from (2)}, \end{cases} \quad x \in \Omega; \tag{10c}$$

$$\varphi_\Omega^{II}(x) \leq \exp[-\beta_0(x_2 - x)/\mu], \quad \text{if } L_\epsilon \text{ from (1)}, \quad x \in \Omega. \tag{10d}$$

Proof: These results follow immediately from the analytic expressions for $\varphi_\Omega^{I,II}(x)$. For the operator L_ϵ from (1) we have

$$\varphi_\Omega^I(x) = \sinh[\beta_0(x_2 - x)/\mu]/\sinh[\beta_0(x_2 - x_1)/\mu],$$

$$\varphi_\Omega^{II}(x) = \sinh[\beta_0(x - x_1)/\mu]/\sinh[\beta_0(x_2 - x_1)/\mu].$$

For the operator L_ϵ from (2) it follows that

$$\varphi_\Omega^I(x) = 1 - \mathcal{I}(x)/\mathcal{I}(x_2), \quad \varphi_\Omega^{II}(x) = 1 - \varphi_\Omega^I(x),$$

$$\mathcal{I}(x) = \int_{x_1}^{x} [E(s)]^{-1} ds, \quad E(x) = \exp\left[\frac{1}{\epsilon}\int_{x_1}^{x} \alpha(s) ds\right].$$

□

We now formulate and prove convergence results for algorithms A1 and A2.

Theorem 1. *If $x_l < x_r$, then the iterative algorithm (3), (4) (that is the Schwarz alternating procedure) converges to the solutions of problems (1) and (2) with rate $0 < q < 1$.*

Proof: We introduce the functions $\zeta^n(x) = v^n(x) - v^{n-1}(x)$, $\xi^n(x) = w^n(x) - w^{n-1}(x)$, $n \geq 2$. From (3), (4) and the mean-value theorem, it follows that $\zeta^n(x)$ and $\xi^n(x)$ are the solutions of the following problems

$$L_\epsilon \zeta^n(x) - f_v^n(x)\zeta^n(x) = 0, \quad x \in \Omega_1, \quad \zeta^n(0) = 0, \quad \zeta^n(x_r) = \xi^{n-1}(x_r), \tag{11a}$$

$$L_\epsilon \xi^n(x) - f_w^n(x)\xi^n(x) = 0, \quad x \in \Omega_2, \quad \xi^n(x_l) = \zeta^n(x_l), \quad \xi^n(1) = 0, \tag{11b}$$

where $f_y^n \equiv f_y(x, \theta_y^n)$, $\theta_y^n \in (y^{n-1}, y^n)$. Let

$$\delta^n = \max[\mid \zeta^n(x_r) \mid, \mid \xi^n(x_l) \mid], \quad n \geq 2.$$

From Lemma 1, we conclude that

$$\max_{x \in \bar{\Omega}_1} \mid \zeta^n(x) \mid \leq \delta^n, \quad \max_{x \in \bar{\Omega}_2} \mid \xi^n(x) \mid \leq \delta^n.$$

Again, using Lemma 1 and (11), it follows that

$$\mid \zeta^n(x_r) \mid = \mid \xi^{n-1}(x_r) \mid \leq \mid \zeta^{n-1}(x_l) \mid \varphi_{\Omega_2}^I(x_r) \leq \mid \zeta^{n-1}(x_r) \mid \varphi_{\Omega_2}^I(x_r)\varphi_{\Omega_1}^{II}(x_l),$$

$$\mid \xi^n(x_l) \mid = \mid \zeta^n(x_l) \mid \leq \mid \xi^{n-1}(x_r) \mid \varphi_{\Omega_1}^{II}(x_l) \leq \mid \xi^{n-1}(x_l) \mid \varphi_{\Omega_2}^I(x_r)\varphi_{\Omega_1}^{II}(x_l).$$

From this we obtain

$$\delta^n \leq q\delta^{n-1}, \quad n \geq 2, \quad q = \varphi_{\Omega_2}^I(x_r)\varphi_{\Omega_1}^{II}(x_l). \tag{12}$$

Using estimate (10a) from Lemma 2, we conclude that $0 < q < 1$. This completes the proof of the convergence of algorithm A1. □

Corollary 1. *For algorithm (3), (4) the following bounds on q hold*

$$q_\mu^{A1} \le \bar{q}_\mu^{A1}, \quad \bar{q}_\mu^{A1} = \exp[-2\beta_0(x_r - x_l)/\mu] < 1,$$

$$q_\epsilon^{A1} \le \bar{q}_\epsilon^{A1}, \quad \bar{q}_\epsilon^{A1} = \exp[-\alpha_0(x_r - x_l)/\epsilon] < 1,$$

where q_μ^{A1}, q_ϵ^{A1} *correspond to problems (1) and (2), respectively.*

Proof: From (12), evaluating q using (10c), (10d) we obtain the required estimates for q_μ^{A1} and q_ϵ^{A1}. □

Theorem 2. *If* $X_l < x_l < x_r < X_r$, *then the iterative algorithm (3), (5) converges to the solutions of problems (1) and (2) with rate* $0 < q < 1$.

Proof: Analogously to the proof of Theorem 1, we introduce the functions $\zeta^n(x) = v^n(x) - v^{n-1}(x)$, $\xi^n(x) = w^n(x) - w^{n-1}(x)$, $\chi^n(x) = z^n(x) - z^{n-1}(x)$, $n \ge 2$. From (3), (5) and the mean-value theorem we conclude that $\zeta^n(x)$, $\xi^n(x)$ and $\chi^n(x)$ are solutions of the problems :

$$L_\epsilon \zeta^n(x) - f_v^n(x)\zeta^n(x) = 0, \quad x \in \Omega_1, \quad \zeta^n(0) = 0, \quad \zeta^n(x_r) = \chi^{n-1}(x_r), \tag{13a}$$

$$L_\epsilon \xi^n(x) - f_w^n(x)\xi^n(x) = 0, \quad x \in \Omega_2, \quad \xi^n(x_l) = \chi^{n-1}(x_l), \quad \xi^n(1) = 0, \tag{13b}$$

$$L_\epsilon \chi^n(x) - f_z^n(x)\chi^n(x) = 0, \quad x \in \omega, \quad \chi^n(X_l) = \zeta^n(X_l), \quad \chi^n(X_r) = \xi^n(X_r). \tag{13c}$$

Let

$$\delta^n = \max[\mid \zeta^n(x_r) \mid, \mid \xi^n(x_l) \mid], \quad n \ge 2.$$

Using the estimates from Lemma 1, we have

$$\max \left\{ \max_{x \in \bar{\Omega}_1} \mid \zeta^n(x) \mid, \max_{x \in \bar{\Omega}_2} \mid \xi^n(x) \mid, \max_{x \in \bar{\omega}} \mid \chi^n(x) \mid \right\} \le \delta^n.$$

By Lemma 1, using the boundary conditions from (13), we conclude that the following estimates hold :

$$\mid \zeta^n(x_r) \mid = \mid \chi^{n-1}(x_r) \mid \le \mid \zeta^{n-1}(x_r) \mid \varphi_{\Omega_1}^{II}(X_l)\varphi_\omega^I(x_r) + \mid \xi^{n-1}(x_l) \mid \varphi_{\Omega_2}^I(X_r)\varphi_\omega^{II}(x_r),$$

$$\mid \xi^n(x_l) \mid = \mid \chi^{n-1}(x_l) \mid \le \mid \zeta^{n-1}(x_r) \mid \varphi_{\Omega_1}^{II}(X_l)\varphi_\omega^I(x_l) + \mid \xi^{n-1}(x_l) \mid \varphi_{\Omega_2}^I(X_r)\varphi_\omega^{II}(x_l). \tag{14}$$

From this, using (10b), we obtain

$$\delta^n \le q\delta^{n-1}, \quad n \ge 2, \quad q = \max\left[\varphi_{\Omega_1}^{II}(X_l), \ \varphi_{\Omega_2}^I(X_r)\right]. \tag{15}$$

Using Lemma 2, it follows that $q < 1$. This proves the convergence of algorithm A2. □

Corollary 2. *For problem (1), we have the following bound on q*

$$q_\mu^{A2} \le \bar{q}_\mu^{A2}, \quad \bar{q}_\mu^{A2} = \max\{\exp[-\beta_0(X_r - x_l)/\mu],\ \exp[-\beta_0(x_r - X_l)/\mu]\}.$$

For problem (2) we have

$$q_\epsilon^{A2} \le \bar{q}_\epsilon^{A2}, \quad \bar{q}_\epsilon^{A2} = 2^{1/2}\max\{\exp[-\alpha_0(X_r - x_l)/2\epsilon],\ \exp[-\alpha_0(x_r - X_l)/2\epsilon]\}.$$

Proof: From (15), evaluating q using the estimates from Lemma 2, we get the required estimate for q_μ^{A2}.
To prove the bound for q_ϵ^{A2}, we express χ^{n-1} in terms of χ^{n-2} and substitute in (14) :

$$\begin{aligned}
|\,\chi^n(X_l)\,| \le \Big\{ &[\,|\,\chi^{n-2}(X_l)\,|\,\varphi_\omega^I(x_r) + |\,\chi^{n-2}(X_r)\,|\,\varphi_\omega^{II}(x_r)\,]\,\varphi_\omega^I(x_r)\varphi_{\Omega_1}^{II}(X_l) + \\
&+ [\,|\,\chi^{n-2}(X_l)\,|\,\varphi_\omega^I(x_l) + |\,\chi^{n-2}(X_r)\,|\,\varphi_\omega^{II}(x_l)\,]\,\varphi_\omega^{II}(x_r)\varphi_{\Omega_2}^I(X_r)\Big\}\,\varphi_{\Omega_1}^{II}(X_l), \\
|\,\chi^n(X_r)\,| \le \Big\{ &[\,|\,\chi^{n-2}(X_l)\,|\,\varphi_\omega^I(x_r) + |\,\chi^{n-2}(X_r)\,|\,\varphi_\omega^{II}(x_r)\,]\,\varphi_\omega^I(x_l)\varphi_{\Omega_1}^{II}(X_l) + \\
&+ [\,|\,\chi^{n-2}(X_l)\,|\,\varphi_\omega^I(x_l) + |\,\chi^{n-2}(X_r)\,|\,\varphi_\omega^{II}(x_l)\,]\,\varphi_\omega^{II}(x_l)\varphi_{\Omega_2}^I(X_r)\Big\}\,\varphi_{\Omega_2}^I(X_r).
\end{aligned}$$

From this we conclude, using Lemma 2, that

$$\begin{aligned}
\delta^n \le \max\Big\{ &\Big[\varphi_\omega^I(x_r)\varphi_{\Omega_1}^{II}(X_l) + \varphi_\omega^{II}(x_r)\varphi_{\Omega_2}^I(X_r)\Big]\,\varphi_{\Omega_1}^{II}(X_l),\ \Big[\varphi_\omega^I(x_l)\varphi_{\Omega_1}^{II}(X_l) + \\
&+\varphi_\omega^{II}(x_l)\varphi_{\Omega_2}^I(X_r)\Big]\,\varphi_{\Omega_2}^I(X_r)\Big\}\,\delta^{n-2} \le \Big[\varphi_\omega^I(x_r) + \varphi_{\Omega_2}^I(X_r)\Big]\,\delta^{n-2}, \quad n \ge 3.
\end{aligned}$$

Thus, it follows that

$$\delta^n \le \Big[\varphi_\omega^I(x_r) + \varphi_{\Omega_2}^I(X_r)\Big]^{1/2}\,\delta^{n-1}, \quad n \ge 2.$$

This proves the bound on q_ϵ^{A2}. □

Remark. Theorems 1 and 2 can be generalized straightforwardly to multi-domain decomposition.

4. Numerical Results

We emphasize here, as is clear from Theorem 2, that the convergence results for algorithm A2 are independent of the singularly perturbed character of problems (1) and (2). To

construct effective numerical methods for algorithm A2, it is necessary to take into account the fact that the solutions of problems (1) and (2) have a boundary layer of size h_ϵ at $x = 0$.

We introduce the "natural" decomposition of the original domain Ω, in which the boundary layer is localized in subdomain Ω_1, and the region where the solution is smooth is included in Ω_2, that is we require:

$$x_l \geq h_\epsilon. \tag{6}$$

Effective numerical methods for singular perturbation problems, such as those based on special nonuniform grids (*cf.* [1, 2]), exhibit the property of uniform convergence with respect to the small parameter. These special grids are constructed in such a way that the number of grid points inside the boundary layers is approximately equal to the number of grid points outside the layers. Thus, if (6) holds and, on subdomains Ω_1, Ω_2 special nonuniform grids are used, the computational cost of the numerical method for problem (3a) on Ω_1 is approximately equal to that for (3b) on Ω_2. This property, known as load balancing, is very important for the implementation of algorithm A2 on parallel computers, since it avoids loss of efficiency due to one processor being idle. The size of the overlap domain $[x_l,\ x_r]$ and hence of the interfacial region $[X_l,\ X_r]$ also affects the cost of an A2-iteration, since the solution of the problem on the interfacial region represents the sequential part of the algorithm A2. It is also worth mentioning, that condition (6) decreases the number of grid points needed for the interfacial problem (5), thus minimizing the time for its solution.

We now present the results of some numerical experiments using iterative algorithms A1 and A2. We shall consider the implementation of these algorithms on a shared memory multiprocessor, the Sequent Balance B21000, in the Department of Mathematics and Computer Science at Kent State University. This has 26 processors, each with a 32-bit National Semiconductor NS32032 capable of 0.75 MIPS, and 32 MB of shared memory. The Balance operating system, DYNIX, provides the ability to bind processes to processors, using the processor affinity facility, and also a utility `team`, which modifies system parameters to permit more accurate timings. The latter gives the highest priority to the program and disables swapping, page fault frequency adjustments and process aging. These privileges allow a user program to execute with a minimum of system overhead to distort benchmark times. This facility was employed in all the timings given below, and effectively eliminated contention for the processors. The coding used the Sequent parallel directives to parallelize do loops in the Fortran code, library calls to the microsecond clock for the timings, and to the parallel processing library to set and manage the number of processors, and to synchronize the processes after the solution of the interfacial problem. The overhead of these operations was negligible.

Example 1. We consider problem (1), where $f(x,u) = 1 - e^{-u},\ u_0 = 1,\ u_1 = 0$. Introduce a non-equidistant grid $\omega_x = \{x_i,\ 0 \leq i \leq N_x\}$. The subdomains Ω_1, Ω_2 and Ω_{inf} from (3), (5a) are chosen such that: $x_l = h_\epsilon = x_j,\ x_r = x_k,\ 0 < j < k < N_x,\ k - j \geq 1,\ X_l =$

$x_{j-1},\ X_r = x_{k+1}$.

In the boundary layer $[0, h_\epsilon]$, the mesh generating function is a logarithmic type function similar to that given in [1]. We approximate the differential equation (1a) by a simple fitted variable-mesh difference formula. The nonlinear algebraic systems (after discretization of (3) and (5)) are solved by a one-step Newton method. In Table 1 and Table 3, we give the number of iterations, K_d, to achieve an error of 10^{-5}, for the direct (undecomposed) method from [1], and for iterative algorithms A1 and A2, K_{A1} and K_{A2} respectively, for various μ and overlapping interval sizes $h = x_r - x_l$. In Table 1 the number of mesh points $N_x = 101$, $j = 51$ and $k \geq 52$ and in Table 3 $N_x = 501$, $j = 251$ and $k \geq 252$. It should be noted that these experiments indicate that the number of iterations is bounded independent of ϵ and is approximately constant for sufficiently small ϵ. Table 2 and Table

	K_d	K_{A1}					K_{A2}				
μ \ h		.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	4	6	10	33	58	202	8	11	26	40	121
.01	4	4	4	4	4	4	4	4	4	5	10
.001	4	4	4	4	4	4	4	4	4	4	4
.0001	4	4	4	4	4	4	4	4	4	4	4

Table 1: Number of iterations for problem (1) for $N_x = 101$

	$S_d = t_d/t_{A2}$					$S_{A1} = t_{A1}/t_{A2}$				
μ \ h	.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	0.71	0.55	0.22	0.15	0.05	1.23	1.59	2.04	3.33	2.94
.01	1.37	1.49	1.59	1.27	0.62	1.61	1.72	1.79	2.17	2.44
.001	1.11	1.23	1.49	1.54	1.56	1.41	1.52	1.72	1.75	1.56
.0001	0.87	0.98	1.28	1.37	1.49	1.20	1.32	1.54	1.61	1.72

Table 2: Speedups S_d and S_{A1} for problem (1) with $N_x = 101$

4 give the speedups $S_d = t_d/t_{A2}$, and $S_{A1} = t_{A1}/t_{A2}$, with respect to the direct method and with respect to algorithm A1. Here t_d is the execution time for the direct method and t_{A1} for algorithm A1 on one processor, and t_{A2} for algorithm A2 on two processors of the Sequent Balance. It should be remarked that in all cases A2 is faster than A1. Note that the dominant effect here is the number of iterations required. In general, one does not expect a two-fold speedup (that is $S = 2$) for A2 over either the direct method or A1 since, due to the interfacial problem, A2 is not perfectly parallelizable. To make an approximate theoretical estimate of the speedup expected, recall that all the problems involved are solutions of tri-diagonal linear systems. Hence the cost is proportional to the number of grid points. On

	K_d	K_{A1}					K_{A2}				
μ \ h		.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	5	6	10	33	58	202	10	15	33	56	145
.01	4	4	4	4	6	14	4	4	5	6	11
.001	4	4	4	4	4	4	4	4	4	4	4
.0001	4	4	4	4	4	4	4	4	4	4	4

Table 3: Number of iterations for problem (1) for $N_x = 501$

	$S_d = t_d/t_{A2}$					$S_{A1} = t_{A1}/t_{A2}$				
μ \ h	.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	0.77	0.54	0.20	0.15	0.06	1.06	1.23	1.49	1.96	2.63
.01	1.54	1.64	1.37	1.14	0.61	1.75	1.85	1.52	1.92	2.44
.001	1.35	1.49	1.67	1.67	1.67	1.61	1.72	1.85	1.85	1.85
.0001	1.06	1.23	1.54	1.59	1.67	1.39	1.52	1.75	1.82	1.85

Table 4: Speedups S_d and S_{A1} for problem (1) with $N_x = 501$

this basis *for a single iteration* $S \leq 100/(50 + 2ninf)$, where *ninf* is the number of points in the interfacial region. This speedup is achieved for the overall time only if $K_{A2} \leq K_d$, and this requires that there be sufficient points in the interfacial region. Thus an optimum strategy is to choose *ninf* as small as possible subject to this requirement. In the case of problem (1) this would give an optimum $S = 1.81$. The remaining degradation seen in the tables can be attributed to the overhead for parallel directives and bus contention.

Example 2. We consider problem (2), where $\alpha = 1 + x$, $f(x, u) = 1 - e^{-u}$, $u_0 = 1$, $u_1 = 0$. We approximate problem (2) by a difference scheme on the special nonuniform grid from [2]. The subdomains Ω_1, Ω_2 and Ω_{inf} are chosen in the same manner as in Example 1. The number of iterations for $N_x = 101$ and $N_x = 501$ are given in Tables 5 and 7, and the corresponding speedups in Tables 6 and 8. We should remark that the anomalous results

	K_d	K_{A1}					K_{A2}				
ϵ \ h		.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	3	7	12	46	81	295	11	17	36	43	51
.01	3	3	3	7	12	43	4	4	6	7	8
.001	3	3	3	3	3	6	3	3	3	3	4
.0001	2	2	2	2	2	2	3	3	3	3	3
.00001	2	2	2	2	2	2	2	2	2	2	2

Table 5: Number of iterations for problem (2) for $N_x = 101$

	$S_d = t_d/t_{A2}$					$S_{A1} = t_{A1}/t_{A2}$				
ϵ \ h	.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	0.43	0.31	0.16	0.13	0.11	1.06	1.23	2.08	3.33	10.00
.01	1.11	1.22	0.87	0.76	0.67	1.27	1.35	2.04	3.03	10.00
.001	1.39	1.52	1.56	1.56	1.15	1.61	1.72	1.72	1.72	2.50
.0001	0.97	1.05	1.08	1.08	1.08	1.11	1.16	1.19	1.19	1.19
.00001	1.35	1.52	1.54	1.54	1.54	1.64	1.61	1.64	1.75	1.75

Table 6: Speedups S_d and S_{A1} for problem (2) with $N_x = 101$

	K_d	K_{A1}					K_{A2}				
ϵ \ h		.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	3	7	12	46	81	294	13	21	64	94	160
.01	3	3	3	7	12	43	4	5	10	14	22
.001	3	3	3	3	3	6	3	3	3	3	4
.0001	2	2	2	2	2	2	3	3	3	3	3
.00001	2	2	2	2	2	2	2	2	2	2	2

Table 7: Number of iterations for problem (2) for $N_x = 501$

	$S_d = t_d/t_{A2}$					$S_{A1} = t_{A1}/t_{A2}$				
ϵ \ h	.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	.38	.27	.10	.07	.03	.91	1.03	1.35	1.61	3.45
.01	1.16	1.03	.58	.43	.28	1.33	1.16	1.35	1.64	3.70
.001	1.47	1.59	1.69	1.67	1.25	1.67	1.79	1.85	1.85	2.78
.0001	1.09	1.18	1.25	1.25	1.25	1.12	1.30	1.35	1.35	1.35
.00001	1.43	1.52	1.61	1.61	1.61	1.61	1.69	1.72	1.75	1.75

Table 8: Speedups S_d and S_{A1} for problem (2) for $N_x = 501$

in Tables 6 and 8 for $\epsilon = .0001$ are due to the fact that $K_{A2} = 3$, whereas $K_d = K_{A1} = 2$. In the case in Table 6, this occurred since, after 2 iterations, A2 had only reduced the error to 9.9×10^{-5}. After 3 iterations it was 6.6×10^{-9}. If we had chosen an error of 5.0×10^{-6} rather than 1.0×10^{-5} this anomaly would not have appeared. This is shown in Tables 9 and 10. In this case, however, a similar problem arises for the case $\epsilon = .00001$. A similar remark holds for the $N_x = 501$ case.

	K_d	K_{A1}					K_{A2}				
ϵ \ h		.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	4	8	13	50	90	341	12	18	39	47	56
.01	3	3	3	8	14	49	4	4	7	8	9
.001	3	3	3	3	3	6	3	3	3	3	4
.0001	3	3	3	3	3	3	3	3	3	3	3
.00001	2	2	2	2	2	2	3	3	3	3	3

Table 9: Number of iterations for problem (2) for $N_x = 101$

	$S_d = t_d/t_{A2}$					$S_{A1} = t_{A1}/t_{A2}$				
ϵ \ h	.1	.05	.01	.005	.001	.1	.05	.01	.005	.001
.1	0.51	0.37	0.18	0.15	0.13	1.11	1.26	2.11	3.38	10.71
.01	1.11	1.21	0.75	0.67	0.60	1.27	1.35	1.99	2.98	9.10
.001	1.40	1.52	1.58	1.57	1.23	1.60	1.70	1.74	1.74	2.48
.0001	1.35	1.45	1.50	1.49	1.49	1.54	1.62	1.66	1.65	1.65
.00001	0.97	1.04	1.08	1.07	1.08	1.10	1.16	1.19	1.18	1.18

Table 10: Speedups S_d and S_{A1} for problem (2) with $N_x = 101$

References

[1] Igor P. Boglaev, A numerical method for a quasilinear singular perturbation problem of elliptic type. U.S.S.R. Comput. Maths. Math. Phys., 28, pp. 492-502, (1988).

[2] Igor P. Boglaev, Numerical method for quasilinear parabolic equation with boundary layer, U.S.S.R. Comput. Maths. Math. Phys., 30, pp. 716-726, (1990).

[3] Igor P. Boglaev, Vadim V. Sirotkin, Domain decomposition technique for singularly perturbed problems and its parallel implementation. In: Proc. 13th IMACS World Congress on Computation and Applied Mathematics, Dublin, (J.J.H. Miller, R. Vichnevetsky, eds.), pp. 522-523, (1991).

[4] Paul A. Farrell, Igor P. Boglaev, Vadim V. Sirotkin, Solution of singular perturbation problems via the domain decomposition method on serial and parallel computers, in Proc. of BAIL 6. Proc. of Sixth International Conference on Boundary and Interior Layers - Computational and Asymptotic Methods (BAIL VI), Front Range Press, Copper Mountain, pp. 92-93, (1992).