Contemporary Mathematics
Volume **157**, 1994

An Iteration Scheme for Non-Symmetric Interface Operator

C.-H. LAI

ABSTRACT. A non-overlapped domain decomposition formulation based on the concept of simple shooting method is presented. A fixed point iteration method without preconditioning of the discretised interface operator is examined. An adaptive parameter based on a simplified version of ϵ-algorithm to accelerate the fixed point iteration is studied.

1. Introduction

In the past couple of years, there has been significant development of domain decomposition preconditioners for conjugate gradient methods. Much of the research has been primarily directed towards finding good preconditioners for symmetric linear systems arising from finite difference or finite element discretisation of elliptic boundary value problems [**1,2**]. There is also some work on non-symmetric problems [**11**] which is again based on domain decomposition preconditioners for conjugate gradient type of iterations. In particular the gradient type of iterations is applied to the entire discretised system, i.e. the domain decomposition is applied at the level of the solution of system of linear equations.

We employ a domain decomposition formulation based on the concept of a simple shooting method and examine a fixed point iteration technique applied to the interfacial system[**6**]. The motivation here is to study a cheap adaptive parameter, initially without preconditioning of the discretised interface operator, to enable acceleration within a fixed point iteration approach. We remark at this stage that the fixed point iteration technique can be used in conjunction with a preconditioner [**5**].

1991 *Mathematics Subject Classification.* Primary 65N55; Secondary 65F10, 39B12.

The author was supported by an ERCIM Fellowship. Travel grant for the DDM6 Conference was provided by SERC, U.K.

The final version of this paper will be submitted for publication elsewhere.

This paper was typeset using $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX Version 2.1.

The basic idea behind a shooting method for the solution of linear 2-point boundary value problems is to assume the derivative of the solution variable at one end of the domain of interest. In terms of difference equations, the idea is to assume a value at the nodal point just in front of the end point, and hence the difference equation can be recurred as far as a second order 2-point boundary value problem is concerned. Based on this calculation, a numerical value at the other end of the domain of interest is obtained. This value is obviously not equal to the boundary condition at the other end. Therefore another approximation at the nodal point just in front of the first end point is assumed and the above recurrence is carried out again to obtain another numerical value at the other end. Finally a linear combination of these two numerical values should equal to the boundary value at the other end point and hence the linear 2-point boundary value problem is solved within two iterations.

One obvious disadvantage of the above method is that the two random assumptions at the nodal point just in front of the first end point can produce a trivial value or a large value at the other end point. Hence a linear combination does not produce the correct answer. One way to tackle this problem is to choose the shooting point at the middle of the domain [**3**], i.e. very large numbers can be avoided by recurring from each end and matching in the middle. The choice of matching in the middle forms the basic idea of our two subdomain approach. Another disadvantage is that the rounding-off error is large whenever a fine grid is used. Hence the numerical solution is only accurate at one end of the domain. One way to tackle this problem is to use multiple shooting method. This idea forms the foundation of our multisubdomain approach.

Our attention is focused on the idea of choosing the shooting point at the middle of the domain in conjunction with a finite difference technique applied to the subdomains separated by the shooting point or the interface as used in domain decomposition methods. Such an idea was initially developed in [**6,9**]. Similar approach for systems of 2-point boundary value problems was reported in [**4**], in which multiple shooting algorithm is applied in a domain decomposition context.

2. Two Subdomain Case

We confine our current study to the following 1-D homogeneous differential equation

$$-\frac{d^2\phi}{dx^2} + c(x)\frac{d\phi}{dx} + r(x)\phi = 0, \;\; a < x < b, \tag{1}$$

subject to Dirichlet boundary conditions $\phi(a) = \phi_a$, $\phi(b) = \phi_b$, $c(x)$ and $r(x)$ are both continuous in $a < x < b$, and $0 < \mathcal{C}_l \leq c(x) \leq \mathcal{C}_u$, $0 \leq \mathcal{R}_l \leq r(x) \leq \mathcal{R}_u$. It is obvious that a unique solution exists for the boundary value problem given in (1). The current approach to the exact solution of the above problem is obtained by considering two related boundary value subproblems as below,

$$(2) \quad -\frac{d^2u_1}{dx^2}+c(x)\frac{du_1}{dx}+r(x)u_1=0,\ a<x<x_I,\ u_1(a)=\phi(a),u_1(x_I)=\lambda,$$

$$(3) \quad -\frac{d^2u_2}{dx^2}+c(x)\frac{du_2}{dx}+r(x)u_2=0,\ x_I<x<b,\ u_2(x_I)=\lambda,u_2(b)=\phi(b).$$

We note that each of the above two subproblems has a unique solution for any value of λ. The coupling at the point x_I for the above subproblems is well known to be (i) the continuity of the function and (ii) the continuity of the derivative of the function at that point. From (2) and (3), the boundary condition imposed at x_I guarantees the continuity of ϕ at x_I. Suppose $u_1=u_1(x;\lambda)$ and $u_2=u_2(x;\lambda)$ denote the solutions of (2) and (3) respectively, then evaluating the solution $\phi(x_I)$ requires a value of λ such that it satisfies the following defect equation,

$$(4) \qquad D(\lambda)\equiv\frac{\partial}{\partial x}u_1(x_I;\lambda)-\frac{\partial}{\partial x}u_2(x_I;\lambda)=0.$$

If $\lambda=\lambda^*$ is a root of the defect equation $D(\lambda)=0$, then the function

$$(5) \qquad \phi(x)=\begin{cases} u_1(x;\lambda^*) & a\le x<x_I \\ \lambda^* & x=x_I \\ u_2(x;\lambda^*) & x_I<x\le b \end{cases},$$

is a solution of the boundary value problem. Note that if $x_I=a$ or $x_I=b$, the above approach reduced to a variant of simple shooting method. In order to approximate the root of $D(\lambda)=0$, we rewrite the equation as $\lambda=G(\lambda)\equiv\lambda-\alpha D(\lambda),\ \alpha\neq 0$, and consider the fixed point iteration scheme

$$(6) \qquad \lambda^{(n+1)}=\lambda^{(n)}-\alpha D(\lambda^{(n)}),\ n=0,1,2,...,$$

where $\lambda^{(0)}$ is arbitrary. Details of the convergence analysis, discretisation error, and matrix representation of the defect equation are given in [8]. Here we only mention two properties related to the iteration scheme (6). It was shown in [8] that $D'(\lambda)>0$ and is bounded and that the scheme (6) converges to the unique root of $D(\lambda)=0$ if α satisfies $0<\alpha<2/\Gamma_u$, where

$$\begin{aligned}\Gamma_u &= ((\mu_u+\sigma_u)e^{\mu_u(x_I-a)}+(\mu_u-\sigma_u)e^{-\mu_u(x_I-a)})/\Delta_{au} \\ &\quad +((\mu_l-\sigma_l)e^{\mu_l(b-x_I)}+(\mu_l+\sigma_l)e^{-\mu_u(b-x_I)})/\Delta_{bl}\end{aligned}$$

$$\begin{aligned}\Delta_{aj} &= e^{\mu_j(x_I-a)}-e^{-\mu_j(x_I-a)}, j=l,u, \\ \Delta_{bj} &= e^{\mu_j(b-x_I)}-e^{-\mu_j(b-x_I)}, j=l,u, \\ \sigma_j &= \mathcal{C}_j/2, \mu_j=\sqrt{\sigma_j^2+\mathcal{R}_j^2}, j=l,u.\end{aligned}$$

However, this result is not very useful in the sense that α is difficult to determine when $D'(\lambda)$ is a dense matrix results from a vector $D(\lambda)$. Some preliminary studies in this respect are given in [7]. Here we extend the idea in [7] and establish in the following theorem an adaptive formula for α based on the scalar ϵ-algorithm [10].

THEOREM 1. *The scalar ϵ-algorithm $\epsilon_{s+1}^{(n)} = \epsilon_{s-1}^{(n+1)} + (\epsilon_s^{(n+1)} - \epsilon_s^{(n)})^{-1}$, $s = 0, 1, \dots$, where $\epsilon_{-1}^{(n)} = 0$, $\epsilon_0^{(n)} = \lambda^{(n)}$, for the acceleration of a converging sequence $\{\lambda^{(n)}\}$, can be imbedded into the iteration scheme* (6) *where α is replaced by α_n and is given by*

$$\alpha_n = \frac{\alpha_{n-1}|D(\lambda^{(n-1)})|}{|D(\lambda^{(n)}) - D(\lambda^{(n-1)})|}, \; n > 0,$$

provided $D'(\lambda) > 0$.

PROOF. For the present case, we only need to construct $\epsilon_2^{(n)}$. From definition, we have $\epsilon_2^{(n)} = \epsilon_0^{(n+1)} + (\epsilon_1^{(n+1)} - \epsilon_1^{(n)})^{-1} = \lambda^{(n+1)} - \frac{\lambda^{(n+1)} - \lambda^{(n)}}{D(\lambda^{(n)}) - D(\lambda^{(n-1)})} D(\lambda^{(n+1)})$ and using the condition $D'(\lambda) > 0$ the result follows.

We apply the above numerical scheme to the model convection diffusion equation $\frac{d^2\phi}{dx^2} - \gamma \frac{d\phi}{dx} = 0$, with boundary conditions $\phi(0) = 0$ and $\phi(1) = 1$ and we choose $x_I = 1 - \frac{1}{\gamma}$. Table 1 shows the number of iterations using an exact solver for the subproblems. Table 2 shows the number of iterations for the case $\gamma = 10$ using a second order finite difference method for the subproblems. In the latter case, we have chosen equal mesh size in both of the subdomains. In both cases, we have used the same $\lambda^{(0)}$ and iterate until $|\lambda^{(n)} - \lambda^{(n-1)}| < 0.5 \times 10^{-5} |\lambda^{(1)} - \lambda^{(0)}|$.

γ	n	$\lambda^{(n)}$	$\phi(x_I)$	$\lvert\phi(x_I) - \lambda^{(n)}\rvert$
10	2	0.36785	0.36785	0.37×10^{-7}
20	2	0.36788	0.36788	0.45×10^{-7}
30	2	0.36788	0.36788	0.81×10^{-7}
40	2	0.36788	0.36788	0.81×10^{-7}
50	2	0.36788	0.36788	0.37×10^{-6}

Table 1 : 1-D model problem using an exact subsolver.

h	n	$\lambda^{(n)}$	$\lvert\phi(x_I) - \lambda^{(n)}\rvert$
0.02	3	0.36662	0.12×10^{-2}
0.01	3	0.36754	0.31×10^{-3}
0.005	3	0.36777	0.76×10^{-4}
0.0025	3	0.36783	0.17×10^{-4}

Table 2 : 1-D model problem using a 2nd order difference subsolver ($\gamma = 10$).

3. Multisubdomain Case

We consider the following $s+1$ related Dirichlet boundary value subproblems,

$$(7) \qquad -\frac{d^2u_k}{dx^2} + c(x)\frac{du_k}{dx} + r(x)u_k = 0, \; u_k(x_{k-1}) = \lambda_{k-1}, u_k(x_k) = \lambda_k,$$

with $x_0 = a$ and $x_{s+1} = b$, $u_1(a) = \phi(a)$ and $u_{s+1}(b) = \phi(b)$. Let $u_k = u_k(x; \lambda)$ denotes the solution of (8) in $x_{k-1} < x < x_k$, where $\lambda = [\lambda_1 \; \lambda_2 \; \dots \; \lambda_s]^T$ is a

s-vector, then evaluating the flux $\phi'(x_k)$, $k = 1, 2, \ldots, s$, requires a vector of λ such that the following vector defect equation is satisfied,

$$D(\lambda) = [D_k(\lambda)] \equiv [\frac{\partial}{\partial x}u_k(x_k;\lambda) - \frac{\partial}{\partial x}u_{k+1}(x_k;\lambda)] = 0. \tag{8}$$

If $\lambda = \lambda^*$ is a root of $D(\lambda) = 0$, then the function

$$\phi(x) = \begin{cases} \lambda^*_{k-1} & x = x_{k-1} \\ u_k(x;\lambda^*) & x_{k-1} < x < x_k, \; k = 1, 2, \ldots, s\,, \\ \lambda^*_k & x = x_k \end{cases} \tag{9}$$

is a solution of the boundary value problem given in (1). We perform numerical experiments to the previous model problem with interfaces chosen as $x_k = \frac{k}{s+1}$. First, we construct the Jacobian matrix $J(\lambda^{(0)}) = D'(\lambda^{(0)})$ numerically and assign $\alpha = J^{-1}$, in which case it is equivalent to a Newton iteration and requires $2s$ subproblem solves. Also each iteration requires $s+1$ subproblem solves in order to compute $D(\lambda^{(n)})$. Since we only need to construct $J(\lambda^{(0)})$ once, the total number of subproblem solves is $(s+1)n_1 + 2s$, where n_1 is the number of iterations. Numerical tests show that $n_1 = 2$ which agrees with the results for 2-subdomain case. Second, we choose $\alpha = (\text{diag}J(\lambda^{(0)}))^{-1}$ which is equivalent to a Jacobi iteration applied to the system $C\lambda = b$ such that $D(\lambda) \equiv C\lambda - b = 0$. The total number of subproblem solves is $(s+1)n_2 + 2s$ where n_2 is the number of iterations. This test is purely for the purpose of comparison with the adaptive α approach. Third, we construct a sequence of scalar parameters $\{\alpha_n\}$ simply by taking the norms of $D(\lambda^{(n-1)})$ and $D(\lambda^{(n)}) - D(\lambda^{(n-1)})$. The amount of work is thus $(s+1)n_3$ subproblem solves. Here we have taken $||.||_2$, however $||.||_\infty$ does not vary the result very much. We present n_2 and n_3 in Table 3 for the previous 1-D model problem using an exact subproblem solver. In all cases, we start from the same $\lambda^{(0)}$ and iterate until $||\lambda^{(n)} - \lambda^{(n-1)}||_\infty \leq 0.5 \times 10^{-5}||\lambda^{(1)} - \lambda^{(0)}||_\infty$.

	$s+1$	4	8	16	32
$\gamma = 10$	n_2	14	45	149	511
	n_3	14	23	37	96
$\gamma = 20$	n_2	8	22	68	236
	n_3	15	21	35	62
$\gamma = 30$	n_2	6	15	44	147
	n_3	15	18	34	58
$\gamma = 40$	n_2	6	12	34	107
	n_3	13	20	37	51
$\gamma = 50$	n_2	4	12	27	85
	n_3	11	20	31	52

Table 3 : Multisubdomain iteration for the 1-D model problem.

The results shown in Table 3 suggests that n_2 becomes unfavourable compared to n_3 as $s+1$ increases. It also suggests that $n_3 \sim 2s$ when $s+1$ is large. We also remark that the approach provides an efficient multiple shooting algorithm for 2-point boundary value problems on parallel computers.

4. A 2-D Non-Symmetric Problem

We consider the following 2-D convection diffusion equation

$$\nabla^2\phi - \gamma\frac{\partial\phi}{\partial x} = 0 \in \Omega = \{(x,y) : 0 < x, y < 1\}, \tag{10}$$

with Dirichlet boundary condition $\phi(0,y) = 0$, $\phi(1,y) = 1$, and $\phi(x,1) = \phi(x,0) = (e^{\gamma x} - 1)/(e^{\gamma} - 1)$. We perform tests with interfaces parallel to x-axis, i.e. the convective flow is tangential to the interface. A second order finite difference scheme is applied to (10) on a regular mesh (x_i, y_j), where $x_i = ih$ and $y_j = jh$, $i, j = 0, 1, \dots, N$, and $h = 1/N$. A total number of $s+1$ subdomains are used and each subdomain is defined as

$$\Omega_k = \{(x,y) : \frac{k-1}{s+1} < y < \frac{k}{s+1},\ 0 < x < 1\},\ k = 1, 2, \dots, s+1, \tag{11}$$

with interfaces $\Gamma_k, k = 1, 2, \dots, s$, defined as

$$\Gamma_k = \partial\Omega_k \cap \partial\Omega_{k+1} = \{0 < x < 1 : y = \frac{k}{s+1}\}. \tag{12}$$

Hence there is a total number of $s(N-1)$ discretised interfacial unknowns. We formulate the vector defect equation of the interface problem and use the adaptive α technique previously described to solve (10) taking $N = 20, 40, 80$, and taking $s+1$ up to $N/2$ or $1/2h$. The numbers of iterations are shown in Tables 4, 5, and 6 respectively.

$s+1$	2	4	10
$\gamma = 10$	31	27	91
$\gamma = 20$	25	25	54
$\gamma = 30$	25	26	37
$\gamma = 40$	30	27	51

Table 4 : 2-D model problem, $N = 20$.

$s+1$	2	4	8	10	20
$\gamma = 10$	30	34	61	65	259
$\gamma = 20$	28	30	43	55	139
$\gamma = 30$	26	32	37	44	73
$\gamma = 40$	29	32	37	40	72
$\gamma = 50$	26	34	38	39	64

Table 5 : 2-D model problem, $N = 40$.

$s+1$	2	4	8	10	20	40
$\gamma = 10$	33	61	79	114	206	838
$\gamma = 20$	28	51	57	67	87	436
$\gamma = 30$	24	38	50	64	87	310
$\gamma = 40$	28	39	50	49	74	219
$\gamma = 50$	26	39	44	49	69	174

Table 6 : 2-D model problem, $N = 80$.

From the above results, one can observe the limited increases in the number of iterations as $s+1$ increases, except for the case $\gamma = 10$, provided $s+1 > 1/2h$. As $s+1 = 1/2h$ the present adaptive α approach becomes unfavourable, however the subproblem solver is extremely cheap and involves only a tridiagonal direct solver of $1/h$ unknowns for each subproblem.

5. Conclusions

A non-overlapped domain decomposition formulation based on the concept of shooting method is presented. The coupling of subdomain solutions is incorporated into a defect equation which involves unknowns along the subdomain interface. A fixed point iteration technique is applied to solve the defect equation. We introduce an adaptive α which can be considered as a simplified version of ϵ-algorithm in order to accelerate the convergence of the iteration scheme. We remark that the present multisubdomain approach provides a fast efficient variant of multiple shooting methods for 2-point boundary value problems on parallel computers.

References

1. J.H. Bramble, J.E. Pasciak, and A.H. Schatz, *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp. **46** (1986), 361–369.
2. T. Chan and D.C. Resasco, *A framework for the analysis and construction of domain decomposition preconditioners*, Domain Decomposition Methods vol. 1, SIAM, Philadelphia, 1988, pp. 217–230.
3. L. Fox, *The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations*, Oxford University Press, London, 1957.
4. H.B. Keller, *Domain decomposition for two-point boundary value problem*, Domain Decomposition Methods vol. 4, SIAM, Philadelphia, 1991, pp. 50–57.
5. Yu.A. Kuznetsov, *Overlapped domain decomposition methods for FE problems with elliptic singular perturbed operators*, Domain Decomposition Methods vol. 4, SIAM, Philadelphia, 1991, pp. 223–241.
6. C.-H. Lai, *Shooting methods for some diffusion and convection problems*, Applied Mathematical Modelling (to appear).
7. ———, *An acceleration technique for a non-overlapped domain decomposition method*, Computational and Applied Mathematics I - Algorithms and Theory, Elsevier, North-Holland, Amsterdam (to appear).
8. ———, *An iteration scheme for non-symmetric interface operator*, ERCIM Research Report (1992).
9. P.-H. Lam, *Parallel Implementation of a Non-Overlapped Domain Decomposition for a Linear 2-Point Boundary Value Problem*, School of Math, Stat, and Comp Report, University of Greenwich, 1991.
10. J.B. McLeod, *A note on the ϵ-algorithm*, Computing **7** (1971), 17–24.
11. G. Meurant, *Incomplete domain decomposition preconditioners for non-symmetric problems*, Domain Decomposition Methods vol. 2, SIAM, Philadelphia, 1989, pp. 219–225.

School of Mathematics, Statistics, and Computing, University of Greenwich, Woolwich Campus, Wellington Street, Woolwich, London SE18 6PF, U.K.

E-mail address: c.h.lai@greenwich.ac.uk