

A PRECONDITIONED CONJUGATE GRADIENT METHOD FOR NONSELFADJOINT OR INDEFINITE ORTHOGONAL SPLINE COLLOCATION PROBLEMS*

RAKHIM AITBAYEV[†] AND BERNARD BIALECKI[‡]

Abstract. We study the computation of the orthogonal spline collocation solution of a linear Dirichlet boundary value problem with a nonselfadjoint or an indefinite operator of the form $Lu = \sum a_{ij}(x)u_{x_i x_j} + \sum b_i(x)u_{x_i} + c(x)u$. We apply a preconditioned conjugate gradient method to the normal system of collocation equations with a preconditioner associated with a separable operator, and prove that the resulting algorithm has a convergence rate independent of the partition step size. We solve a problem with the preconditioner using an efficient direct matrix decomposition algorithm. On a uniform $N \times N$ partition, the cost of the algorithm for computing the collocation solution within a tolerance ϵ is $O(N^2 \ln N |\ln \epsilon|)$.

Key words. nonselfadjoint or indefinite elliptic boundary value problem, orthogonal spline collocation, conjugate gradient method, preconditioner, matrix decomposition algorithm

AMS subject classifications. 65N35, 65N22, 65F10

PII. S0036142901391396

1. Introduction. On $\Omega = (0, 1) \times (0, 1)$ with boundary $\partial\Omega$, we consider the Dirichlet boundary value problem (BVP)

$$(1.1) \quad Lu = f(x), \quad x \in \Omega, \quad u(x) = 0, \quad x \in \partial\Omega,$$

where $x = (x_1, x_2)$ and

$$(1.2) \quad Lu = \sum_{i,j=1}^2 a_{ij}(x) u_{x_i x_j} + \sum_{i=1}^2 b_i(x) u_{x_i} + c(x) u.$$

We assume that a_{ij} , c_i , b , and f are sufficiently smooth, $a_{12}(x) = a_{21}(x)$, $x \in \Omega$, and the a_{ij} satisfy the uniform ellipticity condition

$$(1.3) \quad \nu \sum_{i=1}^2 \eta_i^2 \leq \sum_{i,j=1}^2 a_{ij}(x) \eta_i \eta_j, \quad x \in \Omega, \quad \eta_1, \eta_2 \in R, \quad \nu > 0.$$

In general, the operator L of (1.2) is nonselfadjoint and could be indefinite with respect to the L^2 inner product. The principal part of L is given in nondivergence form rather than the divergence form $\sum_{i,j=1}^2 (a_{ij}(x) u_{x_i})_{x_j}$. While the divergence form is natural for the standard finite element Galerkin method, the nondivergence form is more appropriate for the orthogonal spline collocation (OSC) method since, in this case, the implementation of the OSC method requires neither partial derivatives

*Received by the editors June 25, 2001; accepted for publication (in revised form) November 12, 2002; published electronically May 6, 2003. This work was supported in part by National Science Foundation grant DMS-9805827.

<http://www.siam.org/journals/sinum/41-2/39139.html>

[†]Department of Mathematics, New Mexico Institute of Mining and Technology, Socorro, NM 87801 (aitbayev@nmt.edu).

[‡]Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401 (bbialeck@mines.edu).

of the equation coefficients nor their approximations. Also, in comparison with finite element methods, the OSC method requires no integrals or their approximations to set up the corresponding linear system.

An analysis of OSC for the BVP (1.1)–(1.3) was given in [7], where optimal order L^2 and H^1 error estimates and an optimal H^2 error estimate were obtained. The solution of the resulting linear system by banded Gaussian elimination requires $O(N^4)$ operations on $N \times N$ partition [24, 25, 34]. The application of iterative methods reduces this cost. Classical iterative methods, such as Jacobi, Gauss–Seidel, or SOR, for the OSC solution of Poisson’s equation on a uniform partition were studied in [23, 29, 37]. ADI methods for solving OSC problems with separable operators were investigated in [5, 13, 19].

Since the operator in the BVP is nonseparable, nonselfadjoint, or indefinite, one can attempt to solve the corresponding linear system by preconditioned BICGSTAB, QMR, CGS, and GMRES methods described in [35]. On the other hand, the preconditioned conjugate gradient (PCG) method is an effective method for solving a linear system with a symmetric and positive-definite matrix. We solve the OSC problem by a PCG method applied to a linear system of normal equations. This method is called PCGNR (see section 5.2 in [4] and section 9.5 in [35]).

Preconditioning of a selfadjoint positive-definite operator by a spectrally equivalent operator was suggested by Kantorovich [26]. This idea, used first by D’yakonov [17, 18] for the finite difference solution of a BVP by Richardson’s method, was later extended to the PCG method for the finite element and finite difference solutions of nonselfadjoint or indefinite BVPs [10, 11, 20, 31]. Preconditioning for some nonseparable OSC problems was studied in [6, 27, 28, 38].

Before describing our approach to solving the OSC problem, let us discuss some common techniques used in other discretization methods. Let L_h be a finite difference or a finite element operator associated with a nonselfadjoint or indefinite BVP, and let L_h^* be the adjoint of L_h with respect to an appropriate inner product $(\cdot, \cdot)_h$. Two well-known approaches for solving the equation $L_h u_h = f_h$ are based on preconditioned normal equations:

$$(1.4) \quad L_h^* M_h^{-2} L_h u_h = L_h^* M_h^{-2} f_h,$$

$$(1.5) \quad M_h^{-1} L_h^* M_h^{-1} L_h u_h = M_h^{-1} L_h^* M_h^{-1} f_h,$$

where a selfadjoint and positive-definite operator M_h is a preconditioner for L_h [10, 11, 20, 30, 31]. The operators $L_h^* M_h^{-2} L_h$ and $M_h^{-1} L_h^* M_h^{-1} L_h$ are selfadjoint with respect to $(\cdot, \cdot)_h$ -inner product and M_h -inner product, respectively. Therefore, the equations (1.4) and (1.5) can be solved by the CG method with the corresponding inner products. Analyses of the CG solution of (1.4) and (1.5) are, respectively, related to L^2 -norm and H^1 -norm analyses of a finite difference or a finite element discretization. The finite element equation $L_h u_h = f_h$ can also be solved by modern preconditioned domain decomposition and multilevel methods (see, for example, [33, 36]). However, since these methods are not well developed for OSC, in this article we consider the solution of the OSC problem $L_h u_h = f_h$ approximating BVP (1.1)–(1.2) based on the normal equation

$$(1.6) \quad (M_h^* M_h)^{-1} L_h^* L_h u_h = (M_h^* M_h)^{-1} L_h^* f_h,$$

where a nonselfadjoint or indefinite OSC operator M_h is associated with a separable operator \tilde{L} which is “close” to L . Following an H^2 -norm analysis of [7], we show

that $M_h^* M_h$ and $L_h^* L_h$ are spectrally equivalent with respect to $(\cdot, \cdot)_h$ -inner product. Since the operator $(M_h^* M_h)^{-1} L_h^* L_h$ is selfadjoint and positive-definite with respect to $M_h^* M_h$ -inner product, we solve (1.6) by the corresponding CG method. This method is equivalent to solving the equation $L_h^* L_h u_h = L_h^* f_h$ by PCG with $M_h^* M_h$ as a preconditioner. At each iteration of PCG, a new matrix decomposition algorithm allows us to solve the equation $M_h^* M_h w = r$ in one step rather than in two separate steps $M_h^* z = r$ and $M_h w = z$. On a uniform $N \times N$ partition, the total cost of our PCG algorithm with a tolerance ϵ is $O(N^2 \ln N |\ln \epsilon|)$. The approach presented in this paper was used in [1] for the solution of a nonlinear OSC Dirichlet BVP by Newton's method.

An outline of this paper is as follows. Notation and auxiliary results are introduced in section 2. We prove spectral equivalence of the OSC operators in section 3 and discuss the matrix-vector form of the OSC problem in section 4. In section 5, we prove convergence of the PCG algorithm, and in section 6, we formulate matrix decomposition algorithms for the solution of an equation with the preconditioner. The implementation and the cost are discussed in section 7. In section 8, we present results of our numerical tests, and finally, section 9 is devoted to our conclusions.

2. Preliminaries. For $k = 1, 2$, let $\pi_k = \{x_{k,i}\}_{i=0}^{N_k}$ be a partition of the interval $[0, 1]$ such that

$$0 = x_{k,0} < x_{k,1} < \dots < x_{k,N_k} = 1,$$

and let $h_{k,i} = x_{k,i} - x_{k,i-1}$ for $i = 1, \dots, N_k$. Let

$$\underline{h}_k = \min_i h_{k,i}, \quad \bar{h}_k = \max_i h_{k,i}, \quad h = \max(\bar{h}_1, \bar{h}_2).$$

Throughout we assume that the partitions $\pi_h = \pi_1 \times \pi_2$ are regular; that is, there exist positive constants σ_1, σ_2 , and σ_3 , all independent of h , such that $\sigma_1 \bar{h}_1 \leq \underline{h}_1$, $\sigma_1 \bar{h}_2 \leq \underline{h}_2$, and $\sigma_2 \leq \bar{h}_1 / \bar{h}_2 \leq \sigma_3$.

For an integer $r \geq 3$, let P_r be the set of all polynomials of degree $\leq r$. For $k = 1, 2$, let

$$V_k = \{v \in C^1[0, 1] : v|_{[x_{k,i-1}, x_{k,i}]} \in P_r, \quad i = 1, \dots, N_k\}$$

be the space of Hermite splines of degree r associated with the partition π_k , and let $V_k^0 = \{v \in V_k : v(0) = v(1) = 0\}$. It is easy to verify that the dimension of V_k^0 is $K_k = (r - 1)N_k$. Let $V^0 = V_1^0 \otimes V_2^0$, where \otimes denotes the tensor product of vector spaces. Note that V^0 is the set of all functions that are finite linear combinations of products $v_1(x_1)v_2(x_2)$, where $v_1 \in V_1^0$ and $v_2 \in V_2^0$. The dimension of V^0 is $K = K_1 K_2$.

Let $\{\eta_l\}_{l=1}^{r-1}$ and $\{\omega_l\}_{l=1}^{r-1}$ be, respectively, the nodes and the weights of the $(r - 1)$ -point Gauss quadrature rule on $(0, 1)$. For $k = 1, 2$, let \mathcal{G}_k consist of the points

$$(2.1) \quad \xi_{k,i,l} = x_{k,i-1} + h_{k,i} \eta_l, \quad i = 1, \dots, N_k, \quad l = 1, \dots, r - 1.$$

Then $\mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2$ is the set of Gauss points in Ω associated with the partition π_h . Corollary 5.3 of [32] implies that any $v \in V^0$ is uniquely defined by its values on \mathcal{G} .

For v and z defined on \mathcal{G} , let

$$(2.2) \quad (v, z)_h = \sum_{i=1}^{N_1} h_{1,i} \sum_{k=1}^{r-1} \omega_k \sum_{j=1}^{N_2} h_{2,j} \sum_{l=1}^{r-1} \omega_l (vz)(\xi_{1,i,k}, \xi_{2,j,l})$$

and let $\|v\|_h = \sqrt{(v, v)_h}$. Let $\rho(x)$ be a continuous positive function on $\bar{\Omega}$, and let $\rho_{\min} = \min_{x \in \bar{\Omega}} \rho(x)$ and $\rho_{\max} = \max_{x \in \bar{\Omega}} \rho(x)$. We shall also use

$$(2.3) \quad (v, z)_{h, \rho} = (\rho v, z)_h$$

and $\|v\|_{h, \rho} = \sqrt{(v, v)_{h, \rho}}$. We note that $(\cdot, \cdot)_{h, \rho}$ and $\|\cdot\|_{h, \rho}$ are, respectively, an inner product and a norm in V^0 . It is easy to see that

$$(2.4) \quad \rho_{\min} \|v\|_h \leq \|v\|_{h, \rho} \leq \rho_{\max} \|v\|_h$$

for any v defined on \mathcal{G} .

Throughout, $H^l(\Omega)$ denotes the Sobolev space with the standard norm $\|\cdot\|_{H^l(\Omega)}$ [12]. We write $\partial_k^l = \partial^l / \partial x_k^l$ and $\partial^{(i, j)} = \partial^{i+j} / (\partial x_1^i \partial x_2^j)$. In the following, C denotes a generic positive constant independent of h .

The OSC problem for (1.1)–(1.2) consists of finding $u_h \in V^0$ such that

$$(2.5) \quad Lu_h(\xi) = f(\xi), \quad \xi \in \mathcal{G}.$$

The following result was proved in [7, Theorem 3.3].

THEOREM 2.1. *Let operator L of (1.2) be one-to-one from $\{v \in H^2(\Omega) : v = 0 \text{ on } \partial\Omega\}$ to $L^2(\Omega)$, and let h be sufficiently small. Then the OSC problem (2.5) has a unique solution $u_h \in V^0$. Moreover, if $u \in H^{r+1}(\Omega)$ is the solution of (1.1), then*

$$\|u - u_h\|_{H^2(\Omega)} \leq Ch^{r-1} \|u\|_{H^{r+1}(\Omega)}.$$

3. Spectral equivalence of the OSC operators. The following is the key result of this paper.

THEOREM 3.1. *Let the assumptions of Theorem 2.1 be satisfied. Then there are positive constants γ_1 and γ_2 independent of h such that*

$$(3.1) \quad \gamma_1 \|v\|_{H^2(\Omega)} \leq \|Lv\|_{h, \rho} \leq \gamma_2 \|v\|_{H^2(\Omega)}, \quad v \in V^0.$$

Proof. We note that the inequality

$$C \|v\|_{H^2(\Omega)} \leq \|Lv\|_h + \|v\|_{L^2(\Omega)}, \quad v \in V^0,$$

was proved in [7, (3.20)]. Also, it follows from Lemma 3.2 and (3.21) of [7] that

$$C \|v\|_{L^2(\Omega)} \leq h \|v\|_{H^2(\Omega)} + \|Lv\|_h, \quad v \in V^0.$$

Thus, for h sufficiently small, we have $C \|v\|_{H^2(\Omega)} \leq \|Lv\|_h$, $v \in V^0$, which, along with the first inequality in (2.4), gives the first inequality in (3.1).

Using (1.2) and the boundedness of the coefficients of L , we obtain

$$(3.2) \quad \|Lv\|_h \leq C \sum_{0 \leq i+j \leq 2} \left\| \partial^{(i, j)} v \right\|_h, \quad v \in V^0.$$

Applying the inverse inequality of Theorem 3.2.6 in [12], we have

$$(3.3) \quad \left\| \partial^{(i, j)} v \right\|_h \leq C \left\| \partial^{(i, j)} v \right\|_{L^2(\Omega)}, \quad v \in V^0, \quad 0 \leq i + j \leq 2.$$

Using the second inequality in (2.4), (3.2), (3.3), and the Cauchy–Schwarz inequality, we obtain the second inequality in (3.1). \square

We also consider the separable differential operator

$$(3.4) \quad \tilde{L} = \tilde{L}_1 + \tilde{L}_2,$$

where, for $k = 1, 2$,

$$(3.5) \quad \tilde{L}_k v = \tilde{a}_k(x_k) v_{x_k x_k} + \tilde{b}_k(x_k) v_{x_k} + \tilde{c}_k(x_k) v,$$

\tilde{a}_k, \tilde{b}_k , and \tilde{c}_k are sufficiently smooth, and

$$\tilde{a}_k(x) \geq \nu > 0, \quad x \in \Omega, \quad k = 1, 2.$$

LEMMA 3.1. *Let the assumptions of Theorem 2.1 be satisfied and let the operator \tilde{L} be one-to-one from $\{v \in H^2(\Omega) : v = 0 \text{ on } \partial\Omega\}$ to $L^2(\Omega)$. Then there are positive constants α and β , independent of h , such that*

$$(3.6) \quad \sqrt{\alpha} \|\tilde{L}v\|_{h,\rho} \leq \|Lv\|_{h,\rho} \leq \sqrt{\beta} \|\tilde{L}v\|_{h,\rho}, \quad v \in V^0.$$

Proof. Since \tilde{L} is a special case of L , Theorem 3.1 implies that

$$(3.7) \quad \tilde{\gamma}_1 \|v\|_{H^2(\Omega)} \leq \|\tilde{L}v\|_{h,\rho} \leq \tilde{\gamma}_2 \|v\|_{H^2(\Omega)}, \quad v \in V^0,$$

where the positive constants $\tilde{\gamma}_1$ and $\tilde{\gamma}_2$ are independent of h . Using (3.1) and (3.7), we obtain (3.6) with $\sqrt{\alpha} = \tilde{\gamma}_1/\tilde{\gamma}_2$ and $\sqrt{\beta} = \tilde{\gamma}_2/\tilde{\gamma}_1$. \square

If L_h and M_h are OSC operators from V^0 to V^0 associated with L of (1.2) and \tilde{L} of (3.4)–(3.5), respectively, then (3.6) shows that $L_h^* L_h$ and $M_h^* M_h$ are spectrally equivalent with respect to the inner product $(\cdot, \cdot)_{h,\rho}$. This is equivalent to L_h and M_h being uniformly L^2 -norm equivalent (see (1.15) in [30]). Consequently, our Lemma 3.1 is the OSC counterpart of Lemma 3.1 in [30] for continuous operators.

4. Matrix-vector form of the OSC problem. For $k = 1, 2$, let $\{\phi_{k,j}\}_{j=1}^{K_k}$ be a basis for V_k^0 . Then $\{\phi_j(x)\}_{j=1}^K$, where

$$(4.1) \quad \phi_{K_2(j_1-1)+j_2}(x) = \phi_{1,j_1}(x_1)\phi_{2,j_2}(x_2), \quad j_k = 1, \dots, K_k, \quad k = 1, 2,$$

is a basis for V^0 . Thus, for any $v \in V^0$, there exists a unique vector $[v]_{\mathcal{K}} = [v_1, \dots, v_K]^T \in R^K$ such that

$$(4.2) \quad v(x) = \sum_{j=1}^K v_j \phi_j(x), \quad x \in \bar{\Omega}.$$

Let $\mathcal{G} = \{\xi_i\}_{i=1}^K$, where

$$(4.3) \quad \xi_{(i_1-1)K_2+i_2} = (\xi_{1,i_1}, \xi_{2,i_2}), \quad i_k = 1, \dots, K_k, \quad k = 1, 2,$$

$$(4.4) \quad \xi_{k,(i-1)(r-1)+l} = \xi_{k,i,l}, \quad i = 1, \dots, N_k, \quad l = 1, \dots, r-1,$$

and $\xi_{k,i,l}$ are given by (2.1). For any v defined on \mathcal{G} , we introduce the vector $[v]_{\mathcal{G}} = [v(\xi_1), \dots, v(\xi_K)]^T \in R^K$.

For a matrix A , its (i, j) entry is denoted by $(A)_{ij}$. Let M_L be the matrix defined by

$$(4.5) \quad (M_L)_{ij} = (L\phi_j)(\xi_i), \quad i, j = 1, \dots, K.$$

Then using (4.2) and (4.5), we have

$$(4.6) \quad [Lv]_{\mathcal{G}} = M_L[v]_{\mathcal{H}}, \quad v \in V^0.$$

Let $\rho(x)$ be a continuous positive function on $\bar{\Omega}$. We introduce

$$(4.7) \quad D = \text{diag}(\rho(\xi_1), \dots, \rho(\xi_K)),$$

$$(4.8) \quad W = W_1 \otimes W_2,$$

where \otimes denotes the matrix tensor product and, for $k = 1, 2$,

$$(4.9) \quad W_k = \text{diag}(h_{k,1}, \dots, h_{k,N_k}) \otimes \text{diag}(\omega_1, \dots, \omega_{r-1}).$$

From (2.3), (2.2), (4.3), (4.4), and (4.7)–(4.9), we have

$$(4.10) \quad (v, z)_{h,\rho} = [v]_{\mathcal{G}}^T W D [z]_{\mathcal{G}}.$$

Using (4.6), the OSC problem (2.5) can be rewritten in the matrix-vector form

$$(4.11) \quad M_L[u_h]_{\mathcal{H}} = [f]_{\mathcal{G}}.$$

Multiplying this equation by $M_L^T W D$ on the left, we obtain

$$(4.12) \quad A \vec{u} = \vec{f},$$

where

$$(4.13) \quad A = M_L^T W D M_L, \quad \vec{u} = [u_h]_{\mathcal{H}}, \quad \text{and} \quad \vec{f} = M_L^T W D [f]_{\mathcal{G}}.$$

5. PCG algorithm. Let the operator \tilde{L} be as in (3.4)–(3.5), and let

$$(5.1) \quad \tilde{A} = M_{\tilde{L}}^T W D M_{\tilde{L}},$$

where $M_{\tilde{L}}$ is defined by

$$(5.2) \quad (M_{\tilde{L}})_{ij} = (\tilde{L}\phi_j)(\xi_i), \quad i, j = 1, \dots, K.$$

It follows easily from (4.13) and (5.1) that A and \tilde{A} are symmetric.

LEMMA 5.1. *Let the assumptions of Lemma 3.1 be satisfied. Then the matrices A of (4.13) and \tilde{A} of (5.1) are positive-definite. Moreover,*

$$(5.3) \quad \alpha \vec{v}^T \tilde{A} \vec{v} \leq \vec{v}^T A \vec{v} \leq \beta \vec{v}^T \tilde{A} \vec{v}, \quad \vec{v} \in R^K,$$

where the positive constants α and β are the same as in (3.6).

Proof. Using (4.10), (4.6), and (4.13), we obtain, for $v \in V^0$,

$$(5.4) \quad \|Lv\|_{h,\rho}^2 = (Lv, Lv)_{h,\rho} = [Lv]_{\mathcal{G}}^T W D [Lv]_{\mathcal{G}} = [v]_{\mathcal{H}}^T M_L^T W D M_L [v]_{\mathcal{H}} = [v]_{\mathcal{H}}^T A [v]_{\mathcal{H}}.$$

Hence the first inequality in (3.1) and $\gamma_1 > 0$ imply that A is positive-definite. Similarly, we have $\|\tilde{L}v\|_{h,\rho}^2 = [v]_{\mathcal{H}}^T \tilde{A} [v]_{\mathcal{H}}$. Therefore, (5.3) follows from (3.6) and (5.4).

The second inequality in (5.3) and $\beta > 0$ imply that \tilde{A} is also positive-definite. \square

We solve (4.12) by the PCG method (see Algorithm 9.4.14 in [22]) with \tilde{A} as a preconditioner.

THEOREM 5.1. *Let the assumptions of Lemma 3.1 be satisfied. For an iterate \vec{u}_k generated by the PCG method, let $u_{h,k} \in V^0$ be such that $[u_{h,k}]_{\mathcal{H}} = \vec{u}_k$. Then*

$$(5.5) \quad \|f - Lu_{h,k}\|_{h,\rho} \leq 2\delta^k \|f - Lu_{h,0}\|_{h,\rho}, \quad k = 0, 1, 2, \dots,$$

where $\delta = (\sqrt{\beta/\alpha} - 1)/(\sqrt{\beta/\alpha} + 1)$ and α and β are the same as in (3.6).

Proof. Since A and \tilde{A} are symmetric positive-definite, (5.5) follows from (5.3), Theorem 9.4.14 in [22], (5.4), and (2.5). \square

COROLLARY 5.1. *With δ of Theorem 5.1, we have*

$$(5.6) \quad \|u_h - u_{h,k}\|_{H^2(\Omega)} \leq C\delta^k \|u_h - u_{h,0}\|_{H^2(\Omega)}, \quad k = 0, 1, 2, \dots$$

Proof. Inequality (5.6) follows from (5.5), (2.5), and (3.1). \square

Let $\tilde{r}_k = [f]_{\mathcal{G}} - M_L \vec{u}_k$, $k = 0, 1, \dots$. Then (2.5), (5.4), and (4.11) give $\|\tilde{r}_k\|_{WD} = \|f - Lu_{h,k}\|_{h,\rho}$. Hence, if \tilde{r}_k is required at each iteration and the iterations are terminated when

$$(5.7) \quad \|f - Lu_{h,k}\|_{h,\rho} \leq \epsilon \|f - Lu_{h,0}\|_{h,\rho},$$

then the PCG method can be rewritten in the following form (cf. Algorithm 9.7 in [35]).

ALGORITHM 5.1.

select \vec{u}_0 , $\tilde{r}_0 = [f]_{\mathcal{G}} - M_L \vec{u}_0$, $\vec{r}_0 = M_L^T W D \tilde{r}_0$, solve $\tilde{A} \vec{p}_0 = \vec{r}_0$, $\rho_0 = \vec{r}_0^T \vec{p}_0$,

for $k = 0, 1, 2, \dots$ (as long as $\|\tilde{r}_k\|_{WD} > \epsilon \|\tilde{r}_0\|_{WD}$):

$$\vec{w}_k = M_L \vec{p}_k, \alpha_k = \rho_k / (\vec{w}_k^T W D \vec{w}_k), \vec{u}_{k+1} = \vec{u}_k + \alpha_k \vec{p}_k,$$

$$\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k \vec{w}_k, \vec{r}_{k+1} = M_L^T W D \tilde{r}_{k+1},$$

$$\text{solve } \tilde{A} \vec{z}_{k+1} = \vec{r}_{k+1}, \rho_{k+1} = \vec{r}_{k+1}^T \vec{z}_{k+1}, \vec{p}_{k+1} = \vec{z}_{k+1} + (\rho_{k+1} / \rho_k) \vec{p}_k.$$

6. Preconditioning. At each iteration of Algorithm 5.1, a linear system

$$(6.1) \quad \tilde{A} \vec{w} = \vec{r}$$

must be solved, where \tilde{A} is defined by (5.1)–(5.2). If $\tilde{b}_1 = 0$ or $\tilde{b}_2 = 0$, then (6.1) can be solved by matrix decomposition algorithms which we describe assuming $\tilde{b}_1 = 0$.

For $k = 1, 2$, let I_k be the identity matrix of order K_k , and let the matrices A_k and B_k be defined by

$$(6.2) \quad (A_k)_{ij} = (\tilde{L}_k \phi_{k,j})(\xi_{k,i}), \quad (B_k)_{ij} = \phi_{k,j}(\xi_{k,i}), \quad i, j = 1, 2, \dots, K_k,$$

where \tilde{L}_k is given by (3.5). It follows from (5.2), (3.4), (4.1), (4.3), and (6.2) that

$$(6.3) \quad M_{\tilde{L}} = A_1 \otimes B_2 + B_1 \otimes A_2.$$

With \tilde{a}_1 of (3.5) for $k = 1$, let

$$(6.4) \quad D_1 = \text{diag}(1/\tilde{a}_1(\xi_{1,1}), \dots, 1/\tilde{a}_1(\xi_{1,K_1})).$$

We introduce the $K_1 \times K_1$ matrices

$$(6.5) \quad G = B_1^T W_1 D_1 A_1, \quad F = B_1^T W_1 D_1 B_1,$$

where W_1 is given by (4.9). It was proved in [8, Lemma 3.1] that F is symmetric positive-definite and G is symmetric. Therefore, it follows from [21, Corollary 8.7.2] that there exists a real diagonal matrix

$$(6.6) \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_{K_1})$$

and a real nonsingular matrix Z such that

$$(6.7) \quad Z^T G Z = \Lambda, \quad Z^T F Z = I_1.$$

Now we discuss two approaches to solving (6.1). In the first, we take $\rho = 1$ in $(\cdot, \cdot)_{h,\rho}$ of (2.3) and obtain $D = I$ by (4.7). Hence, by (5.1), the linear system (6.1) becomes

$$M_{\tilde{L}}^T W M_{\tilde{L}} \vec{w} = \vec{r},$$

where the diagonal matrix W is defined by (4.8)–(4.9). Thus, the system in (6.1) can be solved as follows.

ALGORITHM 6.1.

Step 1. Determine Λ and Z satisfying (6.7).

Step 2. Solve $M_{\tilde{L}}^T \vec{z} = \vec{r}$ by a modification of Algorithm I in [8] (see below).

Step 3. Solve the diagonal system $W \vec{v} = \vec{z}$.

Step 4. Solve $M_{\tilde{L}} \vec{w} = \vec{v}$ by Algorithm I in [8].

We note that the matrix decomposition Algorithm I of [8] is based on the decomposition

$$(Z^T B_1^T W_1 D_1 \otimes I_2) M_{\tilde{L}} (Z \otimes I_2) = \Lambda \otimes B_2 + I_1 \otimes A_2,$$

which follows easily from (6.3), (6.5), and (6.7). By taking the transpose of both sides, we also have

$$(Z^T \otimes I_2) M_{\tilde{L}}^T (W_1 D_1 B_1 Z \otimes I_2) = \Lambda \otimes B_2^T + I_1 \otimes A_2^T.$$

Therefore, Step 2 is implemented in a way similar to Step 4 (see [8] for details).

In the second approach to solving (6.1), we take $\rho(x_1, x_2) = 1/\tilde{a}_1(x_1)$, $(x_1, x_2) \in \overline{\Omega}$, which, by (4.7), gives

$$(6.8) \quad D = D_1 \otimes I_2,$$

where D_1 is given by (6.4). Then the system in (6.1) can be solved in one step by a matrix decomposition algorithm which we describe in the following. Since $Z \otimes I_2$ is nonsingular, the system in (6.1) is equivalent to

$$(6.9) \quad S \vec{y} = \vec{d},$$

where $\vec{y} = (Z \otimes I_2)^{-1} \vec{w}$, $\vec{d} = (Z^T \otimes I_2) \vec{r}$, and

$$(6.10) \quad S = (Z^T \otimes I_2) \tilde{A} (Z \otimes I_2).$$

LEMMA 6.1. *Assume that \tilde{L} satisfies the assumptions in Lemma 3.1 and that h is sufficiently small. Then S of (6.10) is a real block diagonal matrix with $K_2 \times K_2$ symmetric positive-definite diagonal blocks*

$$(6.11) \quad S_i = (A_2 + \lambda_i B_2)^T W_2 (A_2 + \lambda_i B_2), \quad i = 1, \dots, K_1.$$

Proof. Using (5.1), (6.3), (4.8), (6.8), (6.5), and $G = G^T$, we obtain

$$(6.12) \quad \begin{aligned} \tilde{A} &= (A_1^T \otimes B_2^T + B_1^T \otimes A_2^T)(W_1 D_1 \otimes W_2)(A_1 \otimes B_2 + B_1 \otimes A_2) \\ &= A_1^T W_1 D_1 A_1 \otimes B_2^T W_2 B_2 + G \otimes B_2^T W_2 A_2 \\ &\quad + G \otimes A_2^T W_2 B_2 + F \otimes A_2^T W_2 A_2. \end{aligned}$$

The second equations in (6.5) and (6.7) give

$$(6.13) \quad B_1 Z Z^T B_1^T W_1 D_1 = I_1.$$

Using (6.13), the first equations in (6.7) and (6.5), and $G^T = G$, we obtain

$$(6.14) \quad Z^T A_1^T W_1 D_1 A_1 Z = Z^T A_1^T W_1 D_1 B_1 Z Z^T B_1^T W_1 D_1 A_1 Z = \Lambda^2.$$

Thus, (6.10), (6.12), (6.14), and (6.7) give

$$(6.15) \quad S = \Lambda^2 \otimes B_2^T W_2 B_2 + \Lambda \otimes B_2^T W_2 A_2 + \Lambda \otimes A_2^T W_2 B_2 + I_1 \otimes A_2^T W_2 A_2.$$

It follows from (6.15) and (6.6) that S is real block diagonal with the diagonal blocks given by (6.11).

We see from (6.11) that each matrix S_i is symmetric and $\vec{v}^T S_i \vec{v} \geq 0$ for any $\vec{v} \in R^{K_2}$. Since Z is nonsingular and \tilde{A} is positive-definite (see Lemma 5.1), it follows from (6.10) that S is nonsingular. This implies that S_i is nonsingular and hence positive-definite. \square

For $\vec{v} \in R^K$, let $[\vec{v}]_i = [v_{(i-1)K_2+1}, \dots, v_{iK_2}]^T \in R^{K_2}$, $i = 1, \dots, K_1$. Based on (6.9) and Lemma 6.1, we can formulate the following matrix decomposition algorithm for the solution of (6.1).

ALGORITHM 6.2.

- Step 1. Determine Λ and Z satisfying (6.7).
- Step 2. Compute $\vec{d} = (Z^T \otimes I_2)\vec{r}$.
- Step 3. Solve $S_i[\vec{y}]_i = [\vec{d}]_i$ for $i = 1, \dots, K_1$.
- Step 4. Compute $\vec{w} = (Z \otimes I_2)\vec{y}$.

7. Implementation and cost. To discuss the implementation and cost, we assume that the basis functions $\{\phi_{k,j}\}_{j=1}^{K_k}$ for V_k^0 , $k = 1, 2$, are B-splines or Hermite-type functions ordered in the standard way. Then matrices A_k and B_k , $k = 1, 2$, in (6.2) are almost block diagonal and have the structure described in [3], depending on the type of basis functions.

Step 1 of Algorithms 6.1 and 6.2 involves solving the symmetric generalized eigenproblem (6.7). This can be done by one of the following three algorithms.

ALGORITHM 7.1.

- Step 1. Compute G and F of (6.5).
- Step 2. Compute band Cholesky factorization $F = LL^T$.
- Step 3. Compute full symmetric $C = L^{-1}GL^{-T}$.
- Step 4. Use QR algorithm to compute the diagonal Λ and an orthogonal Q such that $Q^T C Q = \Lambda$.
- Step 5. Compute $Z = L^{-T}Q$.

ALGORITHM 7.2.

- Step 1. Compute G and F of (6.5).
- Step 2. Compute band Cholesky factorization $F = LL^T$.
- Step 3. Use Crawford's algorithm to compute C and X .
- Step 4. Use band QR algorithm to compute the diagonal Λ and an orthogonal Q such that $Q^T C Q = \Lambda$.
- Step 4. Compute $Z = XQ$.

ALGORITHM 7.3.

- Step 1. Compute full symmetric $C = (W_1 D_1)^{1/2} A_1 B_1^{-1} (W_1 D_1)^{-1/2}$.
- Step 2. Use QR algorithm to compute the diagonal Λ and an orthogonal Q such that $Q^T C Q = \Lambda$.
- Step 3. Compute $Z = B_1^{-1} (W_1 D_1)^{-1/2} Q$.

TABLE 7.1
Costs of Algorithms 7.1–7.3.

Compute	Algorithm 7.1	Algorithm 7.2	Algorithm 7.3
F, G	$O(K_1)$	$O(K_1)$	–
L	$O(K_1)$	$O(K_1)$	–
C (X for Alg. 7.2)	$O(K_1^2)$	$O(K_1^2)$	$O(K_1)$
Λ, Q	$9K_1^3$	$6K_1^3$	$9K_1^3$
Z	$O(K_1^2)$	$2K_1^3$	$O(K_1^2)$
total cost	$9K_1^3 + O(K_1^2)$	$8K_1^3 + O(K_1^2)$	$9K_1^3 + O(K_1^2)$

Algorithm 7.1 is the standard Wilkinson's algorithm (see Algorithm 8.7.1 in [21]). Algorithm 7.2 is based on Crawford's algorithm (see [2] and [14] for details). If $F = LL^T$ is the band Cholesky factorization of F , Crawford's algorithm computes band symmetric C and $X = L^{-T}P$, with P orthogonal, such that $C = X^T GX$ and C is orthogonally similar to Λ . Algorithm 7.3 is based on Step 1 of Algorithm II of [8]. The algorithm uses the factorization $F = LL^T$ with $L = B_1^T(W_1 D_1)^{1/2}$. The costs of Algorithms 7.1–7.3 are given in Table 7.1.

The implementation of Step 4 of Algorithm 6.1 and its cost of $4K_1^2 K_2$ are discussed in [8]. The implementation of Step 2 of Algorithm 6.1 is similar and its cost is also $4K_1^2 K_2$.

Step 2 and Step 4 of Algorithm 6.2 involve K_2 multiplications by $K_1 \times K_1$ matrices Z^T and Z , respectively, and hence each step requires $2K_1^2 K_2$ operations. Each $K_2 \times K_2$ matrix S_i in (6.11) is symmetric, positive-definite, and block tridiagonal with $r - 1$ by $r - 1$ blocks. A linear system with S_i can be solved by a direct block tridiagonal solver (for example, by the routine BLKTRI from the package FISHPACK described in [39]) at the cost $O(K_2)$. Therefore, the cost of Step 3 of Algorithm 6.2 is $O(K_1 K_2)$. Since A_2 and B_2 are almost block diagonal, so is S_i . Hence a linear system with S_i can also be solved by two calls to the routine COLROW [15, 16].

Of course, when Algorithms 6.1 and 6.2 are used in Algorithm 5.1 to solve a linear system with the coefficient matrix \tilde{A} , the matrices Λ and Z are precomputed first. For Algorithm 6.1, the remaining cost is $8K_1^2 K_2$ since this is the cost of all multiplications by Z^T and Z . For Algorithm 6.2, the remaining cost is half of that of Algorithm 6.1.

In a special case of $r = 3$, a uniform partition π_1 , constant coefficients \tilde{a}_1, \tilde{c}_1 of \tilde{L} in (3.5), and $\tilde{b}_1 = 0$, the matrices Λ and Z in (6.7) are known in a closed form. Moreover, it follows from Theorem 2.3 in [9] that matrix Z is given in terms of sines and cosines. Therefore, all multiplications by Z^T and Z in Algorithm 6.1 can be performed using FFTs with the cost $O(K_1 K_2 \log K_1)$. Thus, the total cost of Algorithm 6.1 is $O(K_1 K_2 \log K_1)$. In this case, it follows from (5.6) that the cost of our PCG Algorithm 5.1 with a tolerance ϵ on an $N \times N$ partition is $O(N^2 \ln N |\ln \epsilon|)$.

8. Numerical tests. Before considering our numerical tests, we present an additional result which was used in the tests. Let $\hat{a}_1(x), \hat{a}_2(x)$, and $\hat{c}(x)$ be sufficiently smooth functions on Ω , and for $i = 1, 2$, let $\hat{a}_i(x) \geq \nu > 0, x \in \Omega$. Let

$$(8.1) \quad \hat{L}v = (\hat{a}_1(x)v_{x_1})_{x_1} + (\hat{a}_2(x)v_{x_2})_{x_2} + \hat{c}(x)v.$$

The operator \hat{L} is selfadjoint with respect to the standard L^2 -inner product, and it is negative-definite if $\hat{c}(x) \leq 0, x \in \Omega$. We prove that \hat{L} is indefinite if

$$(8.2) \quad \min_{x \in \Omega} \{\hat{c}(x)\} > 2\pi^2 \max_{x \in \Omega} \{\hat{a}_1(x), \hat{a}_2(x)\}.$$

Using Green’s formula and (8.2), we have, for $v \neq 0$,

$$\begin{aligned}
 \int_{\Omega} \hat{L}v(x) v(x) dx &= - \int_{\Omega} (\hat{a}_1 v_{x_1}^2 + \hat{a}_2 v_{x_2}^2) dx + \int_{\Omega} \hat{c}v^2 dx \\
 &\geq - \max_{x \in \Omega} \{\hat{a}_1(x), \hat{a}_2(x)\} \|\nabla v\|_{L^2(\Omega)}^2 + \min_{x \in \Omega} \{\hat{c}(x)\} \|v\|_{L^2(\Omega)}^2 \\
 (8.3) \quad &> \max_{x \in \Omega} \{\hat{a}_1(x), \hat{a}_2(x)\} \left(2\pi^2 \|v\|_{L^2(\Omega)}^2 - \|\nabla v\|_{L^2(\Omega)}^2 \right),
 \end{aligned}$$

where $\|\nabla v\|_{L^2(\Omega)}^2 = \int_{\Omega} (v_{x_1}^2 + v_{x_2}^2) dx$. It is easy to see that, for

$$v_{k,l}(x) = 2 \sin(k\pi x_1) \sin(l\pi x_2), \quad x \in \Omega,$$

with integers k and l , we have

$$(8.4) \quad \|\nabla v_{k,l}\|_{L^2(\Omega)}^2 = \pi^2(k^2 + l^2) \|v_{k,l}\|_{L^2(\Omega)}^2.$$

Thus, from (8.3) and (8.4), we obtain $\int_{\Omega} \hat{L}v_{1,1}(x) v_{1,1}(x) dx > 0$.

On the other hand, by (8.4), we have

$$\begin{aligned}
 \int_{\Omega} \hat{L}v_{k,l}(x) v_{k,l}(x) dx &\leq -\nu \|\nabla v_{k,l}\|_{L^2(\Omega)}^2 + \max_{x \in \Omega} \{\hat{c}(x)\} \|v_{k,l}\|_{L^2(\Omega)}^2 \\
 &= \left(\max_{x \in \Omega} \{\hat{c}(x)\} - \nu \pi^2(k^2 + l^2) \right) \|v_{k,l}\|_{L^2(\Omega)}^2.
 \end{aligned}$$

Hence, $\int_{\Omega} \hat{L}v_{k,l}(x) v_{k,l}(x) dx < 0$ for sufficiently large k^2 or l^2 . Thus, under the condition (8.2), \hat{L} is indefinite.

Now we describe our numerical tests. The operator L in (1.2) was taken with the coefficients

$$\begin{aligned}
 (8.5) \quad a_{11}(x) &= e^{x_1 x_2}, \quad a_{12}(x) = \alpha / (1 + x_1 + x_2), \quad a_{22}(x) = e^{-x_1 x_2}, \\
 b_1(x) &= x_2 e^{x_1 x_2} + \beta_1 \cos[\pi(x_1 + x_2)], \quad b_2(x) = -x_1 e^{-x_1 x_2} + \beta_2 \sin(2\pi x_1 x_2), \\
 c(x) &= \gamma [1 + 1 / (1 + x_1 + x_2)],
 \end{aligned}$$

where α, β_1, β_2 , and γ are parameters. In BVP (1.1), we set $f(x) = Lu(x)$ for $u(x) = e^{x_1 + x_2} x_1 x_2 (1 - x_1)(1 - x_2)$.

We note that, for the coefficients given by (8.5) with $\alpha = \beta_1 = \beta_2 = 0$, we have $b_1 = (a_{11})_{x_1}$ and $b_2 = (a_{22})_{x_2}$. Therefore, in this case, the operator L in (1.2) can be written in the form of (8.1) with $\hat{a}_i(x) = a_{ii}(x)$, $i = 1, 2$, and $\hat{c}(x) = c(x)$. It follows from (8.5) with $\gamma \geq 0$ that

$$\min_{x \in \Omega} \{c(x)\} = (4/3)\gamma, \quad \max_{x \in \Omega} \{a_{11}(x), a_{22}(x)\} = e.$$

Hence, if $\gamma > (3/2)\pi^2 e \approx 40.243$, then the operator L is indefinite by (8.2).

In our numerical tests, we considered the case of $r = 3$, that is, V^0 is the space of Hermite bicubic splines on a uniform $N \times N$ partition of $\bar{\Omega}$ with the step size $h = 1/N$ (hence, $K_1 = K_2 = 2N$ and $K = 4N^2$). In this case, the standard basis for V^0 is defined as follows. For $k = 1, 2$, let $v_{k,j}, s_{k,j} \in V_k$, $j = 0, \dots, N$, be the “value function” and the “scaled slope function” associated with the node $x_{k,j}$ and defined respectively by

$$v_{k,j}(x_{k,i}) = \delta_{ij}, \quad [v_{k,j}]'(x_{k,i}) = 0, \quad i = 0, \dots, N,$$

TABLE 8.1

Iteration numbers for Algorithm 5.1. $\epsilon = 10^{-10}$ in the stopping condition (5.7). LP = Laplace preconditioner, VCP = variable coefficient preconditioner. 1 = selfadjoint negative-definite L , 2 = selfadjoint indefinite L , 3 = nonselfadjoint L , 4 = general L .

N	1		2		3		4	
	LP	VCP	LP	VCP	LP	VCP	LP	VCP
8	37	22	136	43	133	31	103	59
16	50	26	165	46	163	34	116	68
32	61	30	185	51	173	38	128	75
64	68	33	196	54	178	40	137	81
128	72	34	203	55	184	42	143	84

and

$$s_{k,j}(x_{k,i}) = 0, \quad [s_{k,j}]'(x_{k,i}) = \delta_{ij}/h, \quad i = 0, \dots, N,$$

where δ_{ij} is the Kronecker delta. Then

$$\{\phi_{k,1}, \dots, \phi_{k,K_k}\} = \{s_{k,0}, v_{k,1}, s_{k,1}, \dots, v_{k,N_k-1}, s_{k,N_k-1}, s_{k,N_k}\}$$

is a basis for V_k^0 , $k = 1, 2$, and basis functions for V^0 are given by (4.1).

The OSC problem (2.5) was solved by Algorithm 5.1 with the initial approximation $\vec{u}_0 = \vec{0}$ and the stopping condition (5.7) with $\epsilon = 10^{-10}$. A linear system with a preconditioner was solved by Algorithm 6.2. Since the partition is uniform, Step 1 of Algorithm 6.2 need not be performed, and we used FFTs to implement Steps 2 and 4.

We tested convergence properties of Algorithm 5.1 with two choices of the preconditioner \tilde{A} of (5.1)–(5.2), the first corresponding to $\tilde{L} = \partial^2/\partial x_1^2 + \partial^2/\partial x_2^2$ and the second to the variable coefficient operator \tilde{L} of (3.4)–(3.5) with

$$(8.6) \quad \begin{aligned} \tilde{a}_1(x_1) &= a_{11}(0.5, 0.5), & \tilde{b}_1(x_1) &= 0, & \tilde{c}_1(x_1) &= 0, \\ \tilde{a}_2(x_2) &= a_{22}(0.5, x_2), & \tilde{b}_2(x_2) &= b_2(0.5, x_2), & \tilde{c}_2(x_2) &= c(0.5, x_2). \end{aligned}$$

We refer to these two preconditioners as the Laplacian preconditioner and the variable coefficient preconditioner, respectively. The following cases were tested:

1. selfadjoint negative-definite L ($\alpha = \beta_1 = \beta_2 = \gamma = 0$);
2. selfadjoint indefinite L ($\alpha = \beta_1 = \beta_2 = 0$ and $\gamma = 100$);
3. nonselfadjoint L ($\beta_2 = 100$ and $\alpha = \beta_1 = \gamma = 0$);
4. general L ($\alpha = 0.5$, $\beta_1 = 10$, $\beta_2 = \gamma = 50$).

The numerical results are shown in Table 8.1. We see that PCG with the variable coefficient preconditioner requires fewer iterations than PCG with the Laplacian preconditioner. Moreover, as N increases, the number of PCG iterations grows much slower with the variable coefficient preconditioner than with the Laplacian preconditioner.

In Figure 8.1, we present logarithmic plots of the relative residual curves. The vertical axis represents the values of

$$\log_{10} (\|f - Lu_{h,k}\|_{h,\rho} / \|f\|_{h,\rho}).$$

For both the Laplacian and the variable coefficient preconditioners, we observe monotone convergence. Curve LP1 shows that the Laplacian preconditioner works quite well for the selfadjoint negative-definite problem, but the Laplacian preconditioner

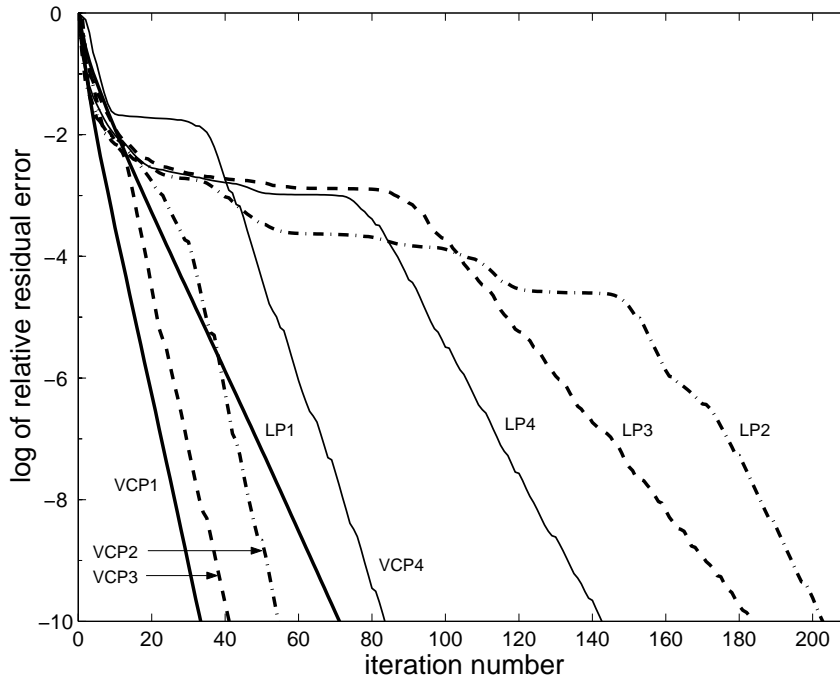


FIG. 8.1. Logarithmic plots of relative residual curves for Algorithm 5.1 ($N = 128$). LP# = Laplacian preconditioner, VCP# = variable coefficient preconditioner. LP1, VCP1 = selfadjoint negative-definite L ; LP2, VCP2 = selfadjoint indefinite L ; LP3, VCP3 = nonselfadjoint L ; LP4, VCP4 = general L .

is not so good for the indefinite, the nonselfadjoint, and the general problems (see the plateaus of curves LP2–LP4). The variable coefficient preconditioner does well in all cases (see curves VCP1–VCP4), although, for the general operator L , the curve VCP4 has a plateau as well. We see that curves VCP1–VCP3 are nearly parallel for large iteration numbers, which indicates that the convergence rates of the PCG with the variable coefficient preconditioner are about the same for different types of L . We observe the same behavior for the Laplacian preconditioner for the selfadjoint and the general problems (see curves LP1, LP2, and LP4). We see from curves LP1 and VCP1 that the convergence of PCG for the selfadjoint negative-definite problem is almost linear. We note that, for the first few iterations, the Laplacian preconditioner works well in all cases; for the general problem, even better than the variable coefficient preconditioner. However, for the larger iteration numbers, curve LP4 has a longer plateau and a smaller slope than curve VCP4.

Next we tested convergence properties of OSC for the general L . Let u_h be the computed OSC solution, and let $e_h = u - u_h$. For any v defined on the partition π_h , let $\|v\|_{\pi_h} = \max_{x \in \pi_h} |v(x)|$. We computed the maximal nodal errors $\|\partial^{(i,j)} e_h\|_{\pi_h}$ for $i, j = 0, 1$ and the Sobolev norms $\|e_h\|_{H^i(\Omega)}$ for $i = 0, 1, 2$, and determined approximate convergence orders using

$$\log_2 \left(\frac{\|\partial^{(i,j)} e_h\|}{\|\partial^{(i,j)} e_{h/2}\|} \right),$$

TABLE 8.2
Maximal nodal errors and convergence orders.

N	e_h		$(e_h)_{x_1}$		$(e_h)_{x_2}$		$(e_h)_{x_1 x_2}$	
4	4.64E-04		5.79E-03		1.34E-02		5.14E-02	
8	9.51E-05	2.29	9.13E-04	2.67	2.26E-03	2.57	9.10E-03	2.50
16	7.63E-07	6.96	6.38E-06	7.16	4.69E-05	5.59	3.79E-04	4.58
32	3.99E-08	4.26	4.47E-07	3.83	3.09E-06	3.92	3.50E-05	3.44
64	2.48E-09	4.01	2.73E-08	4.04	1.93E-07	4.00	3.17E-06	3.46
128	1.55E-10	4.00	1.68E-09	4.02	1.21E-08	4.00	3.21E-07	3.30

TABLE 8.3
Sobolev norm errors and convergence orders.

N	L^2		H^1		H^2	
4	2.11E-04		2.19E-03		4.63E-02	
8	3.56E-04	2.56	3.34E-04	2.72	9.31E-03	2.31
16	2.98E-07	6.90	1.63E-05	4.36	1.72E-03	2.44
32	1.83E-08	4.02	1.88E-06	3.11	3.94E-04	2.12
64	1.13E-09	4.02	2.31E-07	3.03	9.60E-05	2.04
128	7.01E-11	4.01	2.87E-08	3.01	2.38E-05	2.01

where $\|\cdot\|$ is $\|\cdot\|_{\pi_h}$ or $\|\cdot\|_{H^i(\Omega)}$. From the results presented in Table 8.2, we observe the expected order 4 for $\|e_h\|_{\pi_h}$ and the orders 4, 4, and 3 for $\|(e_h)_{x_1}\|_{\pi_h}$, $\|(e_h)_{x_2}\|_{\pi_h}$, and $\|(e_h)_{x_1 x_2}\|_{\pi_h}$, respectively. The last three orders indicate a superconvergence property of OSC. The results in Table 8.3 demonstrate the expected optimal convergence orders for the Sobolev norms.

9. Conclusions. We have shown that PCG is an efficient algorithm for solving the OSC problem (2.5). The convergence analysis of this algorithm is carried out using an H^2 norm analysis of OSC. The convergence rate of PCG is independent of the partition step size h . The approach allows us to use a preconditioner associated with a nonselfadjoint or an indefinite separable operator \tilde{L} . A linear system with the preconditioner can be solved very efficiently using a matrix decomposition algorithm. On a uniform $N \times N$ partition, PCG with a tolerance ϵ requires $O(N^2 \ln N |\ln \epsilon|)$ operations to obtain the Hermite bicubic spline solution of the OSC problem.

Our future work will involve the construction of nonselfadjoint or indefinite OSC domain decomposition and multilevel preconditioners for the OSC problem (2.5).

Acknowledgment. The authors wish to thank Graeme Fairweather for his assistance during the preparation of this paper.

REFERENCES

- [1] A. AITBAYEV, *Orthogonal Spline Collocation for Nonlinear Elliptic Dirichlet Problems*, Ph.D. thesis, University of Kentucky, Lexington, KY, 1998.
- [2] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.
- [3] U. ASCHER, S. PRUESS, AND R. D. RUSSELL, *On spline basis selection for solving differential equations*, SIAM J. Numer. Anal., 20 (1983), pp. 121–142.
- [4] S. F. ASHBY, T. A. MANTEUFFEL, AND P. E. SAYLOR, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568.
- [5] B. BIALECKI, *An alternating direction implicit method for orthogonal spline collocation linear systems*, Numer. Math., 59 (1991), pp. 413–429.

- [6] B. BIALECKI, *Preconditioned Richardson and minimal residual iterative methods for piecewise Hermite bicubic orthogonal spline collocation equations*, SIAM J. Sci. Comput., 15 (1994), pp. 668–680.
- [7] B. BIALECKI, *Convergence analysis of orthogonal spline collocation for elliptic boundary value problems*, SIAM J. Numer. Anal., 35 (1998), pp. 617–631.
- [8] B. BIALECKI AND G. FAIRWEATHER, *Matrix decomposition algorithms in orthogonal spline collocation for separable elliptic boundary value problems*, SIAM J. Sci. Comput., 16 (1995), pp. 330–347.
- [9] B. BIALECKI, G. FAIRWEATHER, AND K. R. BENNETT, *Fast direct solvers for piecewise Hermite bicubic orthogonal spline collocation equations*, SIAM J. Numer. Anal., 29 (1992), pp. 156–173.
- [10] J. H. BRAMBLE, Z. LEYK, AND J. E. PASCIAK, *Iterative schemes for nonsymmetric and indefinite elliptic boundary value problems*, Math. Comp., 60 (1993), pp. 1–22.
- [11] J. H. BRAMBLE AND J. E. PASCIAK, *Preconditioned iterative methods for nonselfadjoint or indefinite elliptic boundary value problems*, in Unification of Finite Element Methods, H. Kardestuncer, ed., North-Holland, Amsterdam, 1984, pp. 167–183.
- [12] P. CIARLET, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
- [13] K. D. COOPER AND P. M. PRENTER, *Alternating direction collocation for separable elliptic partial differential equations*, SIAM J. Numer. Anal., 28 (1991), pp. 711–727.
- [14] C. R. CRAWFORD, *Reduction of a band-symmetric generalized eigenvalue problem*, Comm. ACM, 16 (1973), pp. 41–44.
- [15] J. C. DIAZ, G. FAIRWEATHER, AND P. KEAST, *Algorithm 603 COLROW and ARCECO: FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination*, ACM Trans. Math. Software, 9 (1983), pp. 376–380.
- [16] J. C. DIAZ, G. FAIRWEATHER, AND P. KEAST, *FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination*, ACM Trans. Math. Software, 9 (1983), pp. 358–375.
- [17] Y. G. D'YAKONOV, *An iteration method for solving systems of finite difference equations*, Soviet Math. Dokl., 2 (1961), pp. 647–650.
- [18] Y. G. D'YAKONOV, *The construction of iterative methods based on the use of spectrally equivalent operators*, USSR Comp. Math. and Math. Phys., 6 (1966), pp. 14–46.
- [19] W. R. DYKSEN, *Tensor product generalized ADI methods for separable elliptic problems*, SIAM J. Numer. Anal., 24 (1987), pp. 59–76.
- [20] H. C. ELMAN AND M. H. SCHULTZ, *Preconditioning by fast direct methods for nonself-adjoint nonseparable elliptic equations*, SIAM J. Numer. Anal., 23 (1986), pp. 44–57.
- [21] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The John Hopkins University Press, Baltimore, 1996.
- [22] W. HACKBUSCH, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, New York, 1994.
- [23] A. HADJIDIMOS, E. N. HOUSTIS, J. R. RICE, AND E. VAVALIS, *Modified successive overrelaxation (MSOR) and equivalent 2-step iterative methods for collocation matrices*, J. Comput. Appl. Math., 42 (1992), pp. 375–393.
- [24] E. N. HOUSTIS, W. F. MITCHELL, AND J. R. RICE, *Algorithms INTCOL and HERMCOL: Collocation on rectangular domains with bicubic Hermite polynomials*, ACM Trans. Math. Software, 11 (1985), pp. 416–418.
- [25] E. N. HOUSTIS, W. F. MITCHELL, AND J. R. RICE, *Collocation software for second-order partial differential equations*, ACM Trans. Math. Software, 11 (1985), pp. 379–412.
- [26] L. KANTOROVICH, *Functional analysis and applied mathematics*, Usp. Mat. Nauk, 3 (1948), pp. 89–185 (in Russian).
- [27] H. O. KIM, S. D. KIM, AND Y. H. LEE, *Finite difference preconditioning cubic spline collocation method of elliptic equations*, Numer. Math., 77 (1997), pp. 83–103.
- [28] S. D. KIM AND S. V. PARTER, *Preconditioning cubic spline collocation discretizations of elliptic equations*, Numer. Math., 72 (1995), pp. 39–72.
- [29] Y.-L. LAI, A. HADJIDIMOS, E. N. HOUSTIS, AND J. R. RICE, *On the iterative solution of Hermite collocation equations*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 254–277.
- [30] T. A. MANTEUFFEL AND S. V. PARTER, *Preconditioning and boundary conditions*, SIAM J. Numer. Anal., 27 (1990), pp. 656–694.
- [31] S. V. PARTER AND S.-P. WONG, *Preconditioning second-order elliptic operators: condition numbers and the distribution of the singular values*, J. Sci. Comput., 6 (1991), pp. 129–157.
- [32] P. PERCELL AND M. F. WHEELER, *A C^1 finite element collocation method for elliptic equations*,

- SIAM J. Numer. Anal., 17 (1980), pp. 605–622.
- [33] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Clarendon Press, Oxford, 1999.
 - [34] J. R. RICE AND R. F. BOISVERT, *Solving Elliptic Problems Using ELLPACK*, Springer-Verlag, New York, 1985.
 - [35] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, New York, 1996.
 - [36] B. F. SMITH, P. E. BJØRSTAD, AND W. D. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.
 - [37] W. SUN, *Block iterative algorithms for solving Hermite bicubic collocation equations*, SIAM J. Numer. Anal., 33 (1996), pp. 589–601.
 - [38] W. SUN, W. HUANG, AND R. D. RUSSELL, *Finite difference preconditioning for solving orthogonal collocation equations for boundary value problems*, SIAM J. Numer. Anal., 33 (1996), pp. 2268–2285.
 - [39] P. N. SWARZTRAUBER AND R. SWEET, *Efficient Fortran Subprograms for the Solution of Elliptic Equations*, NCAR TN/IA-109, July 1975.