Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 191 (2003) 601–621
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fast solver for the orthogonal spline collocation solution of the biharmonic Dirichlet problem on rectangles

Bernard Bialecki *

Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, USA

Received 25 October 2002; received in revised form 11 June 2003; accepted 30 June 2003

Abstract

A fast Schur complement algorithm is presented for computing the piecewise Hermite bicubic orthogonal spline collocation solution of the biharmonic Dirichlet problem on a rectangular region. On an $N \times N$ uniform partition, the algorithm, which involves the preconditioned conjugate gradient method and fast Fourier transforms, requires $O(N^2 \log_2 N)$ arithmetic operations.

AMS: 65N35; 65N22

Keywords: Biharmonic Dirichlet problem; Orthogonal spline collocation; Schur complement; Preconditioned conjugate gradient method; Fast Fourier transforms

1. Introduction

This paper is concerned with the solution of the biharmonic Dirichlet problem

$$\Delta^2 u = f \text{ in } \Omega, \quad u = g_1 \text{ on } \partial\Omega, \quad \partial u/\partial n = g_2 \text{ on } \partial\Omega, \tag{1.1}$$

where Δ denotes the Laplacian, $\Omega = (a,b) \times (c,d)$, $\partial\Omega$ is the boundary of Ω, and $\partial/\partial n$ is the outward normal derivative on $\partial\Omega$. Problem (1.1) models bending of a thin elastic rectangular plate, equilibrium of an elastic rectangle, and flow of a viscous fluid in a rectangular cavity (see, e.g., [14]). It can be solved numerically using one of two approaches: direct or mixed. In the first approach, (1.1) is discretized directly using, for example, the finite difference or finite element Galerkin method. In the mixed approach [6], (1.1) is first replaced by a coupled system of two second order differential equations in u and Δu. This system is then discretized using, again, the finite difference or finite element Galerkin method. One advantage of the

* Tel.: +1-303-273-3863; fax: +1-303-273-3875.
E-mail address: bbialeck@mines.edu.

0021-9991/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00342-5

mixed approach is that it produces an approximation not only to u but also to Δu. This is of significant importance in, for example, fluid dynamics, where Δu represents vorticity.

In this paper, we solve (1.1) using the mixed piecewise Hermite bicubic orthogonal spline collocation (OSC) method. Therefore, as in [13,17], we set $v = \Delta u$, introduce a uniform $N \times N$ partition of Ω, and discretize the coupled system of two second order differential equations in u and v using OSC with piecewise Hermite bicubics. The resulting OSC linear system has $4N^2$ unknowns corresponding to the OSC approximation of u and $4N^2$ unknowns corresponding to the OSC approximation of v. This system was solved in [17] at a cost (the number of arithmetic operations) $O(N^3 \log_2 N)$ using a Schur complement method. A capacitance matrix method of [13] has a cost $O(N^3)$. The purpose of this paper is to develop an algorithm of complexity $O(N^2 \log_2 N)$ with a relatively small proportionality constant multiplying $N^2 \log_2 N$. To this end, employing a Schur complement approach, we reduce the OSC linear system to a Schur complement system involving unknowns corresponding to the OSC approximation of v on the two vertical sides of $\partial\Omega$ and to an auxiliary OSC linear system for a biharmonic problem with v, instead of $\partial u/\partial n$, specified on the two vertical sides of $\partial\Omega$. Multiplication of the Schur complement system by an appropriate matrix gives rise to the linear system with a symmetric and positive definite matrix. This new Schur complement system is solved by the preconditioned conjugate gradient (PCG) method with a preconditioner obtained from the auxiliary OSC linear system. Numerical tests indicate that the preconditioner is spectrally equivalent to the Schur complement matrix. Multiplication of a vector by the Schur complement matrix is reduced to computing $8N$ inner products in the space R^{2N} and hence its cost is $32N^2$. The cost of solving a linear system with the preconditioner is $O(N \log_2 N)$. Therefore, with the number of PCG iterations equal to m, the cost of solving the Schur complement system is $32mN^2$. The $4N^2$ unknowns corresponding to the OSC approximation of u are obtained, at a cost $20N^2 \log_2 N$, by solving the auxiliary OSC linear system using separation of variables and fast Fourier transforms. Hence, the nearly optimal total cost of the algorithm to compute the OSC approximation to u is $20N^2 \log_2 N + 32mN^2$. (The additional cost of computing the $4N^2$ unknowns corresponding to the OSC approximation of v is $10N^2 \log_2 N$.) It is worth noting that the term $20N^2 \log_2 N$ is also the cost of the FFT algorithm of [3] for solving the linear system arising from the piecewise Hermite bicubic OSC discretization of Poisson's equation.

An approach similar to the one presented in this paper was used in [4] to compute the Legendre spectral collocation solution of (1.1). However, with polynomials of degree $\leqslant N$ used in both the x- and y-directions, the cost of the corresponding algorithm of [4] is $O(N^3)$. Moreover, in comparison to the present paper, [4] involves the numerical solution of a symmetric eigenvalue problem, a different approach for multiplying a vector by the Schur complement matrix, a special selection of two bases functions to make the preconditioner spectrally equivalent to the Schur complement matrix, and the more expensive solution with the preconditioner.

As expected, numerical results of this paper indicate that the mixed piecewise Hermite bicubic OSC discretization of (1.1) produces fourth order approximations to u in the maximum norm. Moreover, the observed convergence rates are four for the nodal approximations to u_x and u_y. This higher than expected accuracy for the first order derivatives at the partition nodes demonstrates superconvergence phenomena of OSC.

There is an extensive literature on solving (1.1) using the direct or mixed approach with different types of discretizations. Here we mention three additional references on fourth and higher order methods. Reference [1] is concerned with a fourth order direct finite difference method in which a 9-point stencil is used to discretize (1.1) in terms of approximations to u, u_x, and u_y at the partition nodes. A direct finite element method, based on the standard weak form of (1.1), with piecewise Hermite bicubics is considered in [21]. A mixed finite element method with continuous piecewise polynomials is discussed in [10]. In [1,10,21], and in many other papers on finite difference and finite element Galerkin discretizations of (1.1), the resulting linear systems are often solved by iterative methods with multigrid or multilevel (multiplicative or additive) preconditioners. The costs of such iterative multigrid and multilevel multiplicative methods are

$O(N^2|\log_2 \varepsilon|)$, where $0 < \varepsilon < 1$ is the factor by which the error in the initial guess is to be reduced. Since, in general, ε has to be proportion to N^{-k}, with k depending on the order of discretization, the costs of these methods are $O(N^2 \log_2 N)$. Typically, the proportionality constant multiplying $N^2 \log_2 N$ is not given explicitly (see, for example, [1,10,21]). This constant depends on, among other things, the number of smoothings in V- or W-cycle multigrid preconditioners. It is shown in [10] that, in some cases, the number of smoothings must be at least 8 in order to guarantee convergence of the corresponding iterative multigrid method. For these reasons, the proportionality constant multiplying $N^2 \log_2 N$ in many fourth order finite difference and finite element Galerkin methods is expected to be at least as large as a similar constant in the algorithm of this paper. Also, in comparison to iterative multigrid and multilevel multiplicative methods, which are inherently sequential, the present algorithm is well suited for parallel implementation since it consists of independent solutions of linear systems, matrix–vector multiplications, and inner product evaluations.

Our approach of solving numerically (1.1), which is based on finding first an approximation to Δu on the vertical sides of $\partial\Omega$, is particularly well suited for solving plate bending problems [18] with different kinds of clamped and simply supported boundary conditions. For some of these problems our algorithm becomes direct since its PCG component is unnecessary. This is not the case for the iterative multigrid and multilevel methods of [1,10,21].

An outline of this paper is as follows. OSC concepts and necessary results are introduced in Section 2. The OSC biharmonic Dirichlet problem and its matrix–vector form are given in Section 3. The efficient algorithm for solving the OSC linear system is discussed in Section 4. In Section 5 we present numerical results for test problems similar/identical to those in [1,4,11,12,20]. We concentrate, in particular, on the application of our method to the solution of plate bending and fluid flow problems.

2. Preliminaries

For the sake of simplicity, we assume in Sections 2–4 that $\Omega = (0,1) \times (0,1)$ and that N is a power of 2. Let $\{t_n\}_{n=0}^N$ be a uniform partition of $[0,1]$ such that $t_n = nh$, $n = 0,\dots,N$, where $h = 1/N$. Let $\mathcal{M}_h$ be the space of piecewise Hermite cubics on $[0,1]$ defined by

$$\mathcal{M}_h = \{w \in C^1[0,1] : w|_{[t_n,t_{n+1}]} \in P_3,\ n = 0,\dots,N-1\},$$

where P_3 denotes the set of polynomials of degree $\leqslant 3$, and let

$$\mathcal{M}_h^0 = \{w \in \mathcal{M}_h : w(0) = w(1) = 0\}, \quad \mathcal{M}_h^{00} = \{w \in \mathcal{M}_h^0 : w'(0) = w'(1) = 0\}.$$

Let $\{\xi_i\}_{i=1}^{2N}$ be the Gauss points in $(0,1)$ given by

$$\xi_{2n+1} = t_n + h\frac{3-\sqrt{3}}{6}, \quad \xi_{2n+2} = t_n + h\frac{3+\sqrt{3}}{6}, \quad n = 0,\dots,N-1.$$

For $n = 0,\dots,N$, let v_n, $s_n \in \mathcal{M}_h$, associated with t_n, be defined by

$$\begin{aligned} v_n(t_m) &= \delta_{n,m}, & v_n'(t_m) &= 0, \\ s_n(t_m) &= 0, & s_n'(t_m) &= h^{-1}\delta_{n,m}, \end{aligned} \quad n,m = 0,\dots,N, \tag{2.1}$$

where $\delta_{n,m}$ is the Kronecker delta. Then $\{\psi_k\}_{k=0}^{2N+1}$, such that

$$\{\psi_0,\psi_1,\psi_2,\psi_3,\dots,\psi_{2N-2},\psi_{2N-1},\psi_{2N},\psi_{2N+1}\} = \{v_0,s_0,v_1,s_1,\dots,v_{N-1},s_{N-1},s_N,v_N\}, \tag{2.2}$$

is a basis for $\mathcal{M}_h$. Likewise, $\{\psi_k\}_{k=1}^{2N}$ and $\{\psi_k\}_{k=2}^{2N-1}$ are bases for $\mathcal{M}_h^0$ and $\mathcal{M}_h^{00}$, respectively.

We introduce $2N \times 2N$ matrices

$$A = (a_{i,k})_{i,k=1}^{2N}, \quad a_{i,k} = -\psi_k''(\xi_i), \quad B = (b_{i,k})_{i,k=1}^{2N}, \quad b_{i,k} = \psi_k(\xi_i). \tag{2.3}$$

The matrices A and B have the almost block diagonal (ABD) structure [5]

$$A, B = \begin{bmatrix} \times & \times & \times & & & & & & & & \\ \times & \times & \times & & & & & & & & \\ & \times & \times & \times & \times & & & & & & \\ & \times & \times & \times & \times & & & & & & \\ & & & \times & \times & \times & \times & & & & \\ & & & \times & \times & \times & \times & & & & \\ & & & & \ddots & \ddots & \ddots & & & & \\ & & & & & \times & \times & \times & \times & & \\ & & & & & \times & \times & \times & \times & & \\ & & & & & & & \times & \times & \times \\ & & & & & & & \times & \times & \times \end{bmatrix}, \tag{2.4}$$

where $\times$ denotes a nonzero entry.

In addition to A and B of (2.3), we introduce matrices

$$A_t = (a_{i,0}, a_{i,2N+1})_{i=1}^{2N}, \quad B_t = (b_{i,0}, b_{i,2N+1})_{i=1}^{2N}, \tag{2.5}$$

$$A_r = (a_{j,l})_{j=1,l=2}^{2N,2N-1}, \quad B_r = (b_{j,l})_{j=1,l=2}^{2N,2N-1}, \tag{2.6}$$

$$A_e = (a_{j,l})_{j=1,l=0}^{2N,2N+1}, \quad B_e = (b_{j,l})_{j=1,l=0}^{2N,2N+1}, \tag{2.7}$$

where $a_{i,k}$, $b_{i,k}$, $a_{j,l}$, and $b_{j,l}$ are defined as in (2.3). The $2N \times 2$ matrices A_t and B_t of (2.5) have the structure

$$A_t, B_t = \begin{bmatrix} \times & 0 \\ \times & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \times \\ 0 & \times \end{bmatrix}. \tag{2.8}$$

The following remarks follow easily from (2.3)–(2.8).

Remark 2.1. The matrix A_r (B_r) is obtained by deleting the first and last columns from A (B), respectively. The matrix A_e (B_e) is obtained by appending the first and second columns of A_t (B_t) to the left- and right-hand sides of A (B), respectively.

Remark 2.2. The cost of multiplying a vector by B^{T} (or B_r, B_r^{T}, A_r, A_r^{T}) is $\mathrm{O}(N)$.

Eqs. (2.4) and (2.8) imply that $B^{\mathrm{T}}A_t$ and $B^{\mathrm{T}}B_t$ have the structures

$$B^{\mathrm{T}}A_t = \begin{bmatrix} \alpha_1 & 0 \\ \alpha_2 & 0 \\ \alpha_3 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \alpha_{2N-2} \\ 0 & \alpha_{2N-1} \\ 0 & \alpha_{2N} \end{bmatrix}, \quad B^{\mathrm{T}}B_t = \begin{bmatrix} \beta_1 & 0 \\ \beta_2 & 0 \\ \beta_3 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \beta_{2N-2} \\ 0 & \beta_{2N-1} \\ 0 & \beta_{2N} \end{bmatrix}. \tag{2.9}$$

Using the explicit formulas for the entries of B, A_t, and B_t, it is easy to show that

$$\alpha_{2N-2} = \alpha_2, \quad \alpha_{2N-1} = -\alpha_3, \quad \alpha_{2N} = -\alpha_1, \quad \beta_{2N-2} = \beta_2, \quad \beta_{2N-1} = -\beta_3, \quad \beta_{2N} = -\beta_1. \tag{2.10}$$

In [3], explicit formulas were derived for two real matrices: $W = (w_{i,k})_{i,k=1}^{2N}$ and

$$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{2N}), \quad \lambda_k > 0, \ k = 1, \ldots, 2N, \tag{2.11}$$

such that

$$W^{\mathrm{T}}B^{\mathrm{T}}AW = \Lambda, \quad W^{\mathrm{T}}B^{\mathrm{T}}BW = I_{2N}, \tag{2.12}$$

where I_k is the $k \times k$ identity matrix. Since only a different ordering of the same standard basis functions for $\mathscr{M}_h^0$ was used in [3], the matrix W of this paper and the matrix Z of [3, (2.31)] are the same except for the ordering of the rows. In fact, the $(2i)$th row of W is equal to the ith row of Z, $i = 1, \ldots, N-1$, the $(2i-1)$th row of W is equal to the $(N-1+i)$th row of Z, $i = 1, \ldots, N$, and the $(2N)$th rows of W and Z are identical. It also follows from [3, (2.31)] that for $l = 1, \ldots, N$,

$$w_{2N-2,2l} = -w_{2,2l}, \quad w_{2N-2,2l-1} = w_{2,2l-1}, \tag{2.13}$$

$$w_{2N-1,2l} = w_{3,2l}, \quad w_{2N-1,2l-1} = -w_{3,2l-1}, \tag{2.14}$$

$$w_{2N,2l} = w_{1,2l}, \quad w_{2N,2l-1} = -w_{1,2l-1}, \tag{2.15}$$

and that a vector can be multiplied by W (or W^{T}) using one fast sine transform applied to a vector with $N-1$ components and one fast cosine transform applied to a vector with $N+1$ components. Using the results of [19, Sections 4.4.5 and 4.4.6], we obtain the following remark.

Remark 2.3. The cost of multiplying a vector by the matrix W (or W^{T}) is $5N\log_2 N$.

It follows from (2.11) that $\Lambda + \lambda_k I_{2N}$, $k = 1, \ldots, 2N$, is nonsingular, and hence using (2.12) it easy to verify that

$$(A + \lambda_k B)^{-1} = W(\Lambda + \lambda_k I_{2N})^{-1} W^{\mathrm{T}}B^{\mathrm{T}}, \quad k = 1, \ldots, 2N. \tag{2.16}$$

The following result will be needed later.

Lemma 2.1. *If* $U \in \mathscr{M}_h^0 \otimes \mathscr{M}_h^{00}$ *and* $V \in \mathscr{M}_h \otimes \mathscr{M}_h$, *then*

$$\frac{h}{2}\sum_{i,j=1}^{2N}(\Delta UV)(\xi_i, \xi_j) = \frac{h}{2}\sum_{i,j=1}^{2N}(U\Delta V)(\xi_i, \xi_j) + \sum_{j=1}^{2N}(U_x V)(\cdot, \xi_j)\big|_0^1.$$

Proof. Using [9, Lemma 3.1], we have

$$\frac{h}{2}\sum_{i=1}^{2N}(w''z)(\xi_i) = \frac{h}{2}\sum_{i=1}^{2N}(wz'')(\xi_i) + (w'z)|_0^1 - (wz')|_0^1, \quad w,z \in \mathscr{M}_h. \tag{2.17}$$

Hence, (2.17) and $U(\alpha,y)=0$, $\alpha=0,1$, $y\in[0,1]$, give

$$\frac{h}{2}\sum_{i=1}^{2N}(U_{xx}V)(\xi_i,\xi_j) = \frac{h}{2}\sum_{i=1}^{2N}(UV_{xx})(\xi_i,\xi_j) + (U_xV)(\cdot,\xi_j)|_0^1, \quad j=1,\dots,2N.$$

Likewise, (2.17) and $U(x,\beta)=U_y(x,\beta)=0$, $x\in[0,1]$, $\beta=0,1$, give

$$\frac{h}{2}\sum_{j=1}^{2N}(U_{yy}V)(\xi_i,\xi_j) = \frac{h}{2}\sum_{j=1}^{2N}(UV_{yy})(\xi_i,\xi_j), \quad i=1,\dots,2N.$$

Hence the desired result follows from the last two equations. □

3. OSC biharmonic Dirichlet problem and its matrix–vector form

Since the discretization of the nonzero boundary conditions in (1.1) was treated in great detail in [13], we assume, in what follows, that $g_1=g_2=0$. Introducing $v=\Delta u$, we replace (1.1) with the coupled problem

$$-\Delta u + v = 0 \text{ in } \Omega, \quad -\Delta v = -f \text{ in } \Omega, \quad u = \partial u/\partial n = 0 \text{ on } \partial\Omega. \tag{3.1}$$

The piecewise Hermite bicubic OSC problem for (3.1) involves finding $U\in\mathscr{M}_h^{00}\otimes\mathscr{M}_h^{00}$ and $V\in\mathscr{M}_h\otimes\mathscr{M}_h$ such that

$$-\Delta U(\xi_i,\xi_j)+V(\xi_i,\xi_j)=0, \quad -\Delta V(\xi_i,\xi_j) = -f(\xi_i,\xi_j), \quad i,j=1,\dots,2N, \tag{3.2}$$

$$V(\alpha,\beta)=V_y(\alpha,\beta)=0, \quad \alpha,\beta=0,1. \tag{3.3}$$

The scheme (3.2), (3.3), without its analysis, was first considered in [17]. In [13], the existence and uniqueness of U and V were proved in Theorem 5.1 and error bounds for $\|u-U\|_{H^k(\Omega)}$, $k=1,2$, were derived in Theorem 5.2.

Setting

$$u_{1,l}=u_{2N,l}=0, \quad l=2,\dots,2N-1, \tag{3.4}$$

and using the basis functions $\{\psi_k\}_{k=1}^{2N+1}$ of (2.2), we can write

$$U(x,y)=\sum_{k=1}^{2N}\sum_{l=2}^{2N-1}u_{k,l}\psi_k(x)\psi_l(y). \tag{3.5}$$

In a similar way, using (3.3), we have

$$V(x,y)=\sum_{k=1}^{2N}\sum_{l=0}^{2N+1}v_{k,l}\psi_k(x)\psi_l(y)+\sum_{k=0,2N+1}\sum_{l=2}^{2N-1}v_{k,l}\psi_k(x)\psi_l(y). \tag{3.6}$$

Corresponding to (3.5) and (3.6), we introduce vectors

$$\mathbf{u}=[u_{1,2},\dots,u_{1,2N-1},\dots,u_{2N,2},\dots,u_{2N,2N-1}]^{\mathrm{T}}, \tag{3.7}$$

$$\mathbf{u}_{1,*} = [u_{1,2}, \ldots, u_{1,2N-1}]^{\mathrm{T}}, \quad \mathbf{u}_{2N,*} = [u_{2N,2}, \ldots, u_{2N,2N-1}]^{\mathrm{T}}, \tag{3.8}$$

$$\mathbf{v} = [v_{1,0}, \ldots, v_{1,2N+1}, \ldots, v_{2N,0}, \ldots, v_{2N,2N+1}]^{\mathrm{T}}, \tag{3.9}$$

$$\mathbf{v}_{0,*} = [v_{0,2}, \ldots, v_{0,2N-1}]^{\mathrm{T}}, \quad \mathbf{v}_{2N+1,*} = [v_{2N+1,2}, \ldots, v_{2N+1,2N-1}]^{\mathrm{T}}. \tag{3.10}$$

Clearly, $\mathbf{u}_{1,*}$ and $\mathbf{u}_{2N,*}$ of (3.8) are respectively the first and last subvectors of $\mathbf{u}$ in (3.7).

Substituting (3.5) and (3.6) into (3.2), and using (2.3), (2.5)–(2.7), (3.4), we obtain the OSC linear system

$$(A \otimes B_r + B \otimes A_r)\mathbf{u} + (B \otimes B_e)\mathbf{v} + (B_t \otimes B_r)\begin{bmatrix} \mathbf{v}_{0,*} \\ \mathbf{v}_{2N+1,*} \end{bmatrix} = \mathbf{0}, \tag{3.11}$$

$$(A \otimes B_e + B \otimes A_e)\mathbf{v} + (A_t \otimes B_r + B_t \otimes A_r)\begin{bmatrix} \mathbf{v}_{0,*} \\ \mathbf{v}_{2N+1,*} \end{bmatrix} = \mathbf{f}, \tag{3.12}$$

$$-\mathbf{u}_{1,*} = \mathbf{u}_{2N,*} = \mathbf{0}, \tag{3.13}$$

where

$$\mathbf{f} = [f_{1,1}, \ldots, f_{1,2N}, \ldots, f_{2N,1}, \ldots, f_{2N,2N}]^{\mathrm{T}}, \tag{3.14}$$

and $f_{i,j} = -f(\xi_i, \xi_j)$.

4. Algorithm for solving OSC linear system

In this section, we describe an efficient algorithm for solving (3.11)–(3.13).

4.1. Description of the algorithm

Eqs. (3.11)–(3.13) can be written in the compact form

$$S_{11}\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + S_{12}\begin{bmatrix} \mathbf{v}_{0,*} \\ \mathbf{v}_{2N+1,*} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix}, \tag{4.1}$$

$$S_{21}\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{0}, \tag{4.2}$$

where

$$S_{11} = \begin{bmatrix} A \otimes B_r + B \otimes A_r & B \otimes B_e \\ O & A \otimes B_e + B \otimes A_e \end{bmatrix}, \tag{4.3}$$

$$S_{12} = \begin{bmatrix} B_t \otimes B_r \\ A_t \otimes B_r + B_t \otimes A_r \end{bmatrix}, \tag{4.4}$$

$$S_{21} = \begin{bmatrix} -I_{2N-2} & O & O & O \\ O & O & I_{2N-2} & O \end{bmatrix}. \tag{4.5}$$

On comparing (3.13) with (4.2) and (4.5), we obtain the following remark.

Remark 4.1. The computation of $S_{21}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}$ amounts to extracting $-\mathbf{u}_{1,*}$ and $\mathbf{u}_{2N,*}$ from $\mathbf{u}$.

The matrix S_{11} of (4.3) is nonsingular since

$$(A\otimes B_r + B\otimes A_r)\mathbf{u} + (B\otimes B_e)\mathbf{v} = \mathbf{0}, \quad (A\otimes B_e + B\otimes A_e)\mathbf{v} = \mathbf{0},$$

is the matrix–vector form of the following auxiliary OSC problem: find $U\in\mathcal{M}_h^0\otimes\mathcal{M}_h^{00}$, $V\in\mathcal{M}_h^0\otimes\mathcal{M}_h$, such that

$$-\Delta U(\xi_i,\xi_j) + V(\xi_i,\xi_j) = 0, \quad -\Delta V(\xi_i,\xi_j) = 0, \quad i,j = 1,\ldots,2N.$$

It follows from [13, Theorem 4.1] that the only solution to this problem is $U = V = 0$ which gives $\mathbf{u} = \mathbf{v} = \mathbf{0}$.

Using the nonsingularity of S_{11}, we eliminate $\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}$ from (4.1) and (4.2) to obtain

$$S\begin{bmatrix}\mathbf{v}_{0,*}\\ \mathbf{v}_{2N+1,*}\end{bmatrix} = -S_{21}S_{11}^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{f}\end{bmatrix}, \tag{4.6}$$

where the $(4N-4)\times(4N-4)$ Schur complement matrix S is

$$S = -S_{21}S_{11}^{-1}S_{12}. \tag{4.7}$$

The matrix S is nonsingular since it is the Schur complement of the nonsingular S_{11} in the nonsingular $\begin{bmatrix}S_{11} & S_{12}\\ S_{21} & O\end{bmatrix}$. Although the matrix S is nonsymmetric, we have the following key result.

Theorem 4.1. *The matrix*

$$\hat{S} = (I_2\otimes B_r^{\mathrm{T}}B_r)S \tag{4.8}$$

is symmetric and positive definite.

Proof. It follows from the second equation in (2.12) and Remark 2.1 that B_r has linearly independent columns and hence $B_r^{\mathrm{T}}B_r$ is nonsingular. Therefore $\hat{S}$ is nonsingular since it is the product of two nonsingular matrices.

By (4.8) and (4.7), symmetry of $\hat{S}$ is equivalent to

$$-\left((I_2\otimes B_r^{\mathrm{T}}B_r)S_{21}S_{11}^{-1}S_{12}\begin{bmatrix}\mathbf{v}_{0,*}^{(1)}\\ \mathbf{v}_{2N+1,*}^{(1)}\end{bmatrix}, \begin{bmatrix}\mathbf{v}_{0,*}^{(2)}\\ \mathbf{v}_{2N+1,*}^{(2)}\end{bmatrix}\right)_{R^{4N-4}}$$
$$= -\left((I_2\otimes B_r^{\mathrm{T}}B_r)S_{21}S_{11}^{-1}S_{12}\begin{bmatrix}\mathbf{v}_{0,*}^{(2)}\\ \mathbf{v}_{2N+1,*}^{(2)}\end{bmatrix}, \begin{bmatrix}\mathbf{v}_{0,*}^{(1)}\\ \mathbf{v}_{2N+1,*}^{(1)}\end{bmatrix}\right)_{R^{4N-4}} \tag{4.9}$$

for any

$$\mathbf{v}_{0,*}^{(n)} = \left[v_{0,2}^{(n)},\ldots,v_{0,2N-1}^{(n)}\right]^{\mathrm{T}}, \quad \mathbf{v}_{2N+1,*}^{(n)} = \left[v_{2N+1,2}^{(n)},\ldots,v_{2N+1,2N-1}^{(n)}\right]^{\mathrm{T}}, \quad n = 1,2.$$

For $n = 1,2$, let

$$\mathbf{u}^{(n)} = \left[u_{1,2}^{(n)},\ldots,u_{1,2N-1}^{(n)},\ldots,u_{2N,2}^{(n)},\ldots,u_{2N,2N-1}^{(n)}\right]^{\mathrm{T}}$$

and

$$\mathbf{v}^{(n)} = \left[v_{1,0}^{(n)}, \ldots, v_{1,2N+1}^{(n)}, \ldots, v_{2N,0}^{(n)}, \ldots, v_{2N,2N+1}^{(n)}\right]^{\mathrm{T}}$$

be such that

$$S_{11}\begin{bmatrix}\mathbf{u}^{(n)}\\ \mathbf{v}^{(n)}\end{bmatrix} + S_{12}\begin{bmatrix}\mathbf{v}_{0,*}^{(n)}\\ \mathbf{v}_{2N+1,*}^{(n)}\end{bmatrix} = \mathbf{0}. \tag{4.10}$$

Then (4.9) becomes

$$\left((I_2 \otimes B_r^{\mathrm{T}} B_r)S_{21}\begin{bmatrix}\mathbf{u}^{(1)}\\ \mathbf{v}^{(1)}\end{bmatrix}, \begin{bmatrix}\mathbf{v}_{0,*}^{(2)}\\ \mathbf{v}_{2N+1,*}^{(2)}\end{bmatrix}\right)_{R^{4N-4}} = \left((I_2 \otimes B_r^{\mathrm{T}} B_r)S_{21}\begin{bmatrix}\mathbf{u}^{(2)}\\ \mathbf{v}^{(2)}\end{bmatrix}, \begin{bmatrix}\mathbf{v}_{0,*}^{(1)}\\ \mathbf{v}_{2N+1,*}^{(1)}\end{bmatrix}\right)_{R^{4N-4}},$$

which, by (4.5) and Remark 4.1, is the same as

$$\left(B_r\mathbf{u}_{2N,*}^{(1)}, B_r\mathbf{v}_{2N+1,*}^{(2)}\right)_{R^{2N}} - \left(B_r\mathbf{u}_{1,*}^{(1)}, B_r\mathbf{v}_{0,*}^{(2)}\right)_{R^{2N}} = \left(B_r\mathbf{u}_{2N,*}^{(2)}, B_r\mathbf{v}_{2N+1,*}^{(1)}\right)_{R^{2N}} - \left(B_r\mathbf{u}_{1,*}^{(2)}, B_r\mathbf{v}_{0,*}^{(1)}\right)_{R^{2N}}, \tag{4.11}$$

where for $n = 1, 2$,

$$\mathbf{u}_{1,*}^{(n)} = \left[u_{1,2}^{(n)}, \ldots, u_{1,2N-1}^{(n)}\right]^{\mathrm{T}}, \quad \mathbf{u}_{2N,*}^{(n)} = \left[u_{2N,2}^{(n)}, \ldots, u_{2N,2N-1}^{(n)}\right]^{\mathrm{T}}.$$

It follows from (4.3) and (4.4) that (4.10) is the matrix–vector form of the OSC problem

$$-\Delta U^{(n)}(\xi_i, \xi_j) + V^{(n)}(\xi_i, \xi_j) = 0, \quad -\Delta V^{(n)}(\xi_i, \xi_j) = 0, \quad i, j = 1, \ldots, 2N, \tag{4.12}$$

where $U^{(n)}$ and $V^{(n)}$ are given by (3.5) and (3.6), respectively, with $u_{k,l}$ replaced by $u_{k,l}^{(n)}$ and $v_{k,l}$ replaced by $v_{k,l}^{(n)}$. Since $U^{(n)} \in \mathscr{M}_h^0 \otimes \mathscr{M}_h^{00}$ and $V^{(n)} \in \mathscr{M}_h \otimes \mathscr{M}_h$, using (4.12) and Lemma 2.1, we have

$$\frac{h}{2}\sum_{i,j=1}^{2N}(V^{(1)}V^{(2)})(\xi_i, \xi_j) = \frac{h}{2}\sum_{i,j=1}^{2N}(\Delta U^{(1)}V^{(2)})(\xi_i, \xi_j) = \sum_{j=1}^{2N}(U_x^{(1)}V^{(2)})(\cdot, \xi_j)|_0^1.$$

Likewise,

$$\frac{h}{2}\sum_{i,j=1}^{2N}(V^{(1)}V^{(2)})(\xi_i, \xi_j) = \frac{h}{2}\sum_{i,j=1}^{2N}(V^{(1)}\Delta U^{(2)})(\xi_i, \xi_j) = \sum_{j=1}^{2N}(U_x^{(2)}V^{(1)})(\cdot, \xi_j)|_0^1,$$

and hence

$$h\sum_{j=1}^{2N}(U_x^{(1)}V^{(2)})(\cdot, \xi_j)|_0^1 = h\sum_{j=1}^{2N}(U_x^{(2)}V^{(1)})(\cdot, \xi_j)|_0^1. \tag{4.13}$$

Using representations of $U^{(n)}$ and $V^{(n)}$ (cf. (3.5) and (3.6)), (2.2), (2.1), and (2.6), it is easy to verify that (4.13) is the same as (4.11). This proves symmetry of $\hat{S}$.

Finally we show that $\hat{S}$ is positive definite. Using the first part of the proof with $\mathbf{v}_{0,*}^{(2)} = \mathbf{v}_{0,*}^{(1)}$ and $\mathbf{v}_{2N+1,*}^{(2)} = \mathbf{v}_{2N+1,*}^{(1)}$, we have

$$\begin{aligned}\left(\hat{S}\begin{bmatrix}\mathbf{v}_{0,*}^{(1)}\\ \mathbf{v}_{2N+1,*}^{(1)}\end{bmatrix}, \begin{bmatrix}\mathbf{v}_{0,*}^{(1)}\\ \mathbf{v}_{2N+1,*}^{(1)}\end{bmatrix}\right)_{R^{4N-4}} &= \left(B_r\mathbf{u}_{2N,*}^{(1)}, B_r\mathbf{v}_{2N+1,*}^{(1)}\right)_{R^{2N}} - \left(B_r\mathbf{u}_{1,*}^{(1)}, B_r\mathbf{v}_{0,*}^{(1)}\right)_{R^{2N}}\\ &= h\sum_{j=1}^{2N}(U_x^{(1)}V^{(1)})(\cdot, \xi_j)|_0^1 = \frac{h^2}{2}\sum_{i,j=1}^{2N}(V^{(1)}V^{(1)})(\xi_i, \xi_j),\end{aligned}$$

which shows that $\hat{S}$ is nonnegative definite. Since $\hat{S}$ is also symmetric and nonsingular, this implies that $\hat{S}$ is positive definite. □

Based on Theorem 4.1, (4.6), and (4.1), we arrive at the following algorithm for solving (3.11)–(3.13).

Algorithm I
Step 1. Compute $\mathbf{r} = -S_{21}S_{11}^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{f}\end{bmatrix}$.
Step 2. Compute $\hat{\mathbf{r}} = (I_2 \otimes B_r^{\mathrm{T}}B_r)\mathbf{r}$ and solve $\hat{S}\begin{bmatrix}\mathbf{v}_{0,*}\\ _{2N+1,*}\end{bmatrix} = \hat{\mathbf{r}}$ for $\mathbf{v}_{0,*}$ and $\mathbf{v}_{2N+1,*}$.
Step 3. Compute $\mathbf{u}$ and $\mathbf{v}$ given by $\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} = S_{11}^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{f}\end{bmatrix} - S_{11}^{-1}S_{12}\begin{bmatrix}\mathbf{v}_{0,*}\\ \mathbf{v}_{2N+1,*}\end{bmatrix}$.

In Sections 4.2 and 4.3, we explain how to solve linear systems with matrices S_{11} and $\hat{S}$, and in Section 4.4 we describe a final implementation of Algorithm I and give its cost.

4.2. Solving a linear system with matrix S_{11}

In addition to vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{f}$ of the forms (3.7), (3.9), and (3.14), respectively, we introduce

$$\mathbf{g} = [g_{1,1}, \ldots, g_{1,2N}, \ldots, g_{2N,1}, \ldots, g_{2N,2N}]^{\mathrm{T}}. \tag{4.14}$$

Then it follows from (4.3) that

$$S_{11}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{g}\\ \mathbf{f}\end{bmatrix} \tag{4.15}$$

can be rewritten as

$$(A \otimes B_r + B \otimes A_r)\mathbf{u} + (B \otimes B_e)\mathbf{v} = \mathbf{g}, \quad (A \otimes B_e + B \otimes A_e)\mathbf{v} = \mathbf{f}. \tag{4.16}$$

For future reference, in place of (4.16), we consider

$$\begin{aligned}&(A \otimes B + B \otimes A)\mathbf{u}_e + (B \otimes B_e)\mathbf{v} = \mathbf{g},\\ &(A \otimes B_e + B \otimes A_e)\mathbf{v} = \mathbf{f},\\ &-\mathbf{u}_{*,1} = \mathbf{a}, \quad \mathbf{u}_{*,2N} = \mathbf{b},\end{aligned} \tag{4.17}$$

where

$$\mathbf{u}_e = [u_{1,1}, \ldots, u_{1,2N}, \ldots, u_{2N,1}, \ldots, u_{2N,2N}]^{\mathrm{T}}, \tag{4.18}$$

$$\mathbf{u}_{*,1} = [u_{1,1}, \ldots, u_{2N,1}]^{\mathrm{T}}, \quad \mathbf{u}_{*,2N} = [u_{1,2N}, \ldots, u_{2N,2N}]^{\mathrm{T}}, \tag{4.19}$$

and

$$\mathbf{a} = [a_1, \ldots, a_{2N}]^{\mathrm{T}}, \quad \mathbf{b} = [b_1, \ldots, b_{2N}]^{\mathrm{T}}. \tag{4.20}$$

The vector $\mathbf{u}_e$ in (4.18) is obtained by appending the components of $\mathbf{u}_{*,1}$ and $\mathbf{u}_{*,2N}$ in (4.19) to $\mathbf{u}$ of (3.7). Hence, (4.17) arises from (4.16) by replacing $\mathbf{u}$, A_r, and B_r with $\mathbf{u}_e$, A, and B, respectively, and by including two additional equations for $\mathbf{u}_{*,1}$ and $\mathbf{u}_{*,2N}$.

Clearly, nonsingularity of S_{11} implies that of the matrix in the linear system (4.17).

Remark 4.2. The linear system (4.15) is a special case of (4.17) with $\mathbf{a} = \mathbf{b} = \mathbf{0}$.

In the remainder of this section, we discuss a matrix decomposition algorithm, based on (2.12), for solving (4.17).

It follows from the second equation in (2.12) that W^TB^T is nonsingular and that $W^{-1} = W^TB^TB$. Hence (4.17) is equivalent to

$$\begin{aligned}&(W^TB^T \otimes I_{2N})(A \otimes B + B \otimes A)(W \otimes I_{2N})\mathbf{u}'_e + (W^TB^T \otimes I_{2N})(B \otimes B_e)(W \otimes I_{2N+2})\mathbf{v}' = \mathbf{g}',\\ &(W^TB^T \otimes I_{2N})(A \otimes B_e + B \otimes A_e)(W \otimes I_{2N+2})\mathbf{v}' = \mathbf{f}',\\ &-\mathbf{u}'_{*,1} = \mathbf{a}', \quad \mathbf{u}'_{*,2N} = \mathbf{b}',\end{aligned} \tag{4.21}$$

where $\mathbf{u}'_e$, $\mathbf{v}'$, $\mathbf{u}'_{*,1}$, and $\mathbf{u}'_{*,2N}$ are such that

$$\mathbf{u}_e = (W \otimes I_{2N})\mathbf{u}'_e, \quad \mathbf{v} = (W \otimes I_{2N+2})\mathbf{v}', \quad \mathbf{u}_{*,1} = W\mathbf{u}'_{*,1}, \quad \mathbf{u}_{*,2N} = W\mathbf{u}'_{*,2N}, \tag{4.22}$$

and

$$\mathbf{g}' = (W^TB^T \otimes I_{2N})\mathbf{g}, \quad \mathbf{f}' = (W^TB^T \otimes I_{2N})\mathbf{f}, \quad \mathbf{a}' = W^TB^TB\mathbf{a}, \quad \mathbf{b}' = W^TB^TB\mathbf{b}. \tag{4.23}$$

The vectors $\mathbf{u}'_e$, $\mathbf{v}'$, $\mathbf{u}'_{*,1}$, $\mathbf{u}'_{*,2N}$, $\mathbf{g}'$, $\mathbf{f}'$, and $\mathbf{a}'$, $\mathbf{b}'$ have the same forms as $\mathbf{u}_e$ of (4.18), $\mathbf{v}$ of (3.9), $\mathbf{u}_{*,1}$, $\mathbf{u}_{*,2N}$ of (4.19), $\mathbf{g}$ of (4.14), $\mathbf{f}$ of (3.14), and $\mathbf{a}$, $\mathbf{b}$ of (4.20), respectively. In what follows, we denote the components of the primed vectors by the corresponding primed letters, for example,

$$\mathbf{u}'_e = [u'_{1,1}, \ldots, u'_{1,2N}, \ldots, u'_{2N,1}, \ldots, u'_{2N,2N}]^T. \tag{4.24}$$

Using (4.21) and (2.12), we have

$$\begin{aligned}&(\Lambda \otimes B + I_{2N} \otimes A)\mathbf{u}'_e + (I_{2N} \otimes B_e)\mathbf{v}' = \mathbf{g}',\\ &(\Lambda \otimes B_e + I_{2N} \otimes A_e)\mathbf{v}' = \mathbf{f}',\\ &\mathbf{u}'_{*,1} = -\mathbf{a}', \quad \mathbf{u}'_{*,2N} = \mathbf{b}'.\end{aligned} \tag{4.25}$$

For $k = 1, \ldots, 2N$, we introduce

$$\mathbf{u}'_{k,*} = [u'_{k,1}, \ldots, u'_{k,2N}]^T, \quad \mathbf{v}'_{k,*} = [v'_{k,0}, \ldots, v'_{k,2N+1}]^T, \tag{4.26}$$

and

$$\mathbf{g}'_{k,*} = [g'_{k,1}, \ldots, g'_{k,2N}]^T, \quad \mathbf{f}'_{k,*} = [f'_{k,1}, \ldots, f'_{k,2N}]^T. \tag{4.27}$$

With the use of (2.11), (4.25) becomes

$$(A + \lambda_k B)\mathbf{u}'_{k,*} + B_e\mathbf{v}'_{k,*} = \mathbf{g}'_{k,*}, \quad (A_e + \lambda_k B_e)\mathbf{v}'_{k,*} = \mathbf{f}'_{k,*}, \quad u'_{k,1} = -a'_k, \quad u'_{k,2N} = b'_k, \tag{4.28}$$

where $k = 1, \ldots, 2N$. Since $\lambda_k > 0$ (see (2.11)), the unique solvability of (4.28) is guaranteed by [13, Theorem 2.1]. By Remark 2.1, (4.28) can be rewritten in the form

$$R_{11}^{(k)}\begin{bmatrix}\mathbf{u}'_{k,*}\\ \mathbf{v}''_{k,*}\end{bmatrix} + R_{12}^{(k)}\begin{bmatrix}v'_{k,0}\\ v'_{k,2N+1}\end{bmatrix} = \begin{bmatrix}\mathbf{g}'_{k,*}\\ \mathbf{f}'_{k,*}\end{bmatrix}, \tag{4.29}$$

$$R_{21}\begin{bmatrix}\mathbf{u}'_{k,*}\\ \mathbf{v}''_{k,*}\end{bmatrix} = \begin{bmatrix}-a'_k\\ b'_k\end{bmatrix}, \tag{4.30}$$

where $\mathbf{v}''_{k,*} = [v'_{k,1}, \ldots, v'_{k,2N}]^T$,

$$R_{11}^{(k)} = \begin{bmatrix}A + \lambda_k B & B\\ O & A + \lambda_k B\end{bmatrix}, \quad R_{12}^{(k)} = \begin{bmatrix}B_t\\ A_t + \lambda_k B_t\end{bmatrix}, \quad R_{21} = \begin{bmatrix}10\ldots000\ldots0\\ 00\ldots010\ldots0\end{bmatrix}, \tag{4.31}$$

and where the 1 in the second row of R_{21} appears in the column $2N$. Nonsingularity of $A+\lambda_k B$ (cf. (2.16)) implies that of $R_{11}^{(k)}$. Solving (4.29) for $\begin{bmatrix}\mathbf{u}'_{k,*}\\ \mathbf{v}''_{k,*}\end{bmatrix}$ and substituting it into (4.30), we obtain

$$\begin{bmatrix} v'_{k,0} \\ v'_{k,2N+1} \end{bmatrix} = [R^{(k)}]^{-1}\begin{bmatrix} a'_k \\ -b'_k \end{bmatrix} + \tilde{R}^{(k)}\begin{bmatrix} \mathbf{g}'_{k,*} \\ \mathbf{f}'_{k,*} \end{bmatrix}, \tag{4.32}$$

where the 2×2 matrix

$$R^{(k)} = R_{21}[R_{11}^{(k)}]^{-1}R_{12}^{(k)}, \tag{4.33}$$

and the $2\times 4N$ matrix

$$\tilde{R}^{(k)} = [R^{(k)}]^{-1}R_{21}[R_{11}^{(k)}]^{-1} = \begin{bmatrix} \left[\mathbf{q}_1^{(k)}\right]^{\mathrm{T}} & \left[\mathbf{q}_2^{(k)}\right]^{\mathrm{T}} \\ \left[\mathbf{q}_3^{(k)}\right]^{\mathrm{T}} & \left[\mathbf{q}_4^{(k)}\right]^{\mathrm{T}} \end{bmatrix}, \tag{4.34}$$

where $\mathbf{q}_i^{(k)} \in R^{2N}$, $i = 1,2,3,4$. For each $k = 1,\ldots,2N$, the $2\times 4N$ matrix $R_{21}[R_{11}^{(k)}]^{-1}$ is precomputed at a cost $\mathrm{O}(N)$ using COLROW [7,8] for solving ABD systems. Then $R^{(k)}$ of (4.33) and $\tilde{R}^{(k)}$ of (4.34) are precomputed at costs $\mathrm{O}(1)$ and $\mathrm{O}(N)$, respectively. To solve (4.28), we first compute $v'_{k,0}$, $v'_{k,2N+1}$ at a cost $\mathrm{O}(N)$ using (4.32), and then use COLROW to solve (4.29) for $\mathbf{u}'_{k,*}$, $\mathbf{v}''_{k,*}$ at a cost $\mathrm{O}(N)$.

We arrive at the following algorithm for solving (4.17).

Algorithm II

Step 1. Compute $\mathbf{g}'$, $\mathbf{f}'$, $\mathbf{a}'$, and $\mathbf{b}'$ using (4.23).
Step 2. For $k = 1,\ldots,2N$, solve (4.28) for $\mathbf{u}'_{k,*}$ and $\mathbf{v}'_{k,*}$ using (4.32) and (4.29).
Step 3. Compute $\mathbf{u}_e$ and $\mathbf{v}$ using (4.22).

Using Remarks 2.2 and 2.3 we obtain the following remark.

Remark 4.3. In Step 1 of Algorithm II, the cost of computing $\mathbf{g}'$, $\mathbf{f}'$ is $10N^2\log_2 N$ each, and the cost of computing $\mathbf{a}'$, $\mathbf{b}'$ is $\mathrm{O}(N\log_2 N)$ each. The cost of Step 2 to compute the solutions $\mathbf{u}'_{k,*}$, $\mathbf{v}'_{k,*}$, $k = 1,\ldots,2N$, of (4.28) is $\mathrm{O}(N^2)$. In Step 3, the cost of computing $\mathbf{u}_e$ and $\mathbf{v}$ is $10N^2\log_2 N$ each.

4.3. Solving a linear system with matrix $\hat{S}$

It follows from Theorem 4.1 that the PCG method can be used to perform the second part of Step 2 of Algorithm I. Therefore, in this section, we discuss multiplication of a vector by $\hat{S}$, selection of a preconditioner for $\hat{S}$, and the solution of a linear system with the preconditioner.

Let $\mathbf{v}_{0,*}$ and $\mathbf{v}_{2N+1,*}$ be arbitrary vectors of the forms (3.10). Then (4.8), (4.7), and Remark 4.1 imply that in order to compute $\hat{S}\begin{bmatrix}\mathbf{v}_{0,*}\\ \mathbf{v}_{2N,*}\end{bmatrix}$, we have to determine

$$\begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} = S_{12}\begin{bmatrix} \mathbf{v}_{0,*} \\ \mathbf{v}_{2N+1,*} \end{bmatrix}, \tag{4.35}$$

solve (4.15) for $\mathbf{u}_{1,*}$, $\mathbf{u}_{2N,*}$, and finally calculate $B_r^{\mathrm{T}}B_r\mathbf{u}_{1,*}$, $B_r^{\mathrm{T}}B_r\mathbf{u}_{2N,*}$. It follows from Remark 4.2, (4.35), (4.4), (4.23), (2.9), (4.27), (2.10), and (2.13)–(2.15) that for $l = 1,\ldots,N$,

$$\mathbf{g}'_{2l,*} = \delta_{2l}B_r(\mathbf{v}_{0,*} - \mathbf{v}_{2N+1,*}), \quad \mathbf{g}'_{2l-1,*} = \delta_{2l-1}B_r(\mathbf{v}_{0,*} + \mathbf{v}_{2N+1,*}), \tag{4.36}$$

$$\mathbf{f}'_{2l,*} = (\gamma_{2l}B_r + \delta_{2l}A_r)(\mathbf{v}_{0,*} - \mathbf{v}_{2N+1,*}), \quad \mathbf{f}'_{2l-1,*} = (\gamma_{2l-1}B_r + \delta_{2l-1}A_r)(\mathbf{v}_{0,*} + \mathbf{v}_{2N+1,*}), \tag{4.37}$$

where

$$\delta_k = \sum_{i=1}^{3} \beta_i w_{i,k}, \quad \gamma_k = \sum_{i=1}^{3} \alpha_i w_{i,k}, \quad k = 1, \ldots, 2N.$$

Using (4.29) and (4.31), we obtain for $k = 1, \ldots, 2N$,

$$\mathbf{u}'_{k,*} = (A + \lambda_k B)^{-1}[\mathbf{g}'_{k,*} - B(A + \lambda_k B)^{-1}\mathbf{f}'_{k,*}] + v'_{k,0}\mathbf{p}_1^{(k)} + v'_{k,2N+1}\mathbf{p}_2^{(k)}, \tag{4.38}$$

where $\mathbf{p}_1^{(k)}, \mathbf{p}_2^{(k)} \in R^{2N}$ are such that

$$\left[\,\mathbf{p}_1^{(k)}, \mathbf{p}_2^{(k)}\,\right] = (A + \lambda_k B)^{-1}[B(A + \lambda_k B)^{-1}(A_t + \lambda_k B_t) - B_t]. \tag{4.39}$$

Since $\mathbf{a}' = \mathbf{b}' = \mathbf{0}$ by (4.23), (4.32), (4.34), (4.36), and (4.37) imply that for $l = 1, \ldots, N$,

$$\begin{aligned} v'_{2l,0} &= (\mathbf{z}_1^{(2l)}, \mathbf{v}_{0,*} - \mathbf{v}_{2N+1,*})_{R^{2N-2}}, \quad v'_{2l,2N+1} = (\mathbf{z}_2^{(2l)}, \mathbf{v}_{0,*} - \mathbf{v}_{2N+1,*})_{R^{2N-2}}, \\ v'_{2l-1,0} &= (\mathbf{z}_1^{(2l-1)}, \mathbf{v}_{0,*} + \mathbf{v}_{2N+1,*})_{R^{2N-2}}, \quad v'_{2l-1,2N+1} = (\mathbf{z}_2^{(2l-1)}, \mathbf{v}_{0,*} + \mathbf{v}_{2N+1,*})_{R^{2N-2}}, \end{aligned} \tag{4.40}$$

where

$$\mathbf{z}_1^{(k)} = B_r^{\mathrm{T}}\left[\delta_k \mathbf{q}_1^{(k)} + \gamma_k \mathbf{q}_2^{(k)}\right] + \delta_k A_r^{\mathrm{T}} \mathbf{q}_2^{(k)}, \quad \mathbf{z}_2^{(k)} = B_r^{\mathrm{T}}\left[\delta_k \mathbf{q}_3^{(k)} + \gamma_k \mathbf{q}_4^{(k)}\right] + \delta_k A_r^{\mathrm{T}} \mathbf{q}_4^{(k)}. \tag{4.41}$$

It follows from (3.8), (4.18), (4.22), (4.24), (4.26), (4.28) with $a'_k = b'_k = 0$, and (2.15) that

$$\begin{bmatrix} 0 \\ \mathbf{u}_{1,*} \\ 0 \end{bmatrix} = \sum_{l=1}^{N} w_{1,2l}\mathbf{u}'_{2l,*} + \sum_{l=1}^{N} w_{1,2l-1}\mathbf{u}'_{2l-1,*}, \tag{4.42}$$

$$\begin{bmatrix} 0 \\ \mathbf{u}_{2N,*} \\ 0 \end{bmatrix} = \sum_{l=1}^{N} w_{1,2l}\mathbf{u}'_{2l,*} - \sum_{l=1}^{N} w_{1,2l-1}\mathbf{u}'_{2l-1,*}. \tag{4.43}$$

Using (4.38), (4.36), (4.37), (2.16), and (2.12), we obtain

$$\begin{aligned} \sum_{l=1}^{N} w_{1,2l}\mathbf{u}'_{2l,*} &= WD_1W^{\mathrm{T}}B^{\mathrm{T}}B_r(\mathbf{v}_{0,*} - \mathbf{v}_{2N+1,*}) - WD_2W^{\mathrm{T}}B^{\mathrm{T}}A_r(\mathbf{v}_{0,*} - \mathbf{v}_{2N+1,*}) \\ &\quad + \sum_{l=1}^{N}\left[w_{1,2l}v'_{2l,0}\mathbf{p}_1^{(2l)} + w_{1,2l}v'_{2l,2N+1}\mathbf{p}_2^{(2l)}\right], \end{aligned} \tag{4.44}$$

$$\begin{aligned} \sum_{l=1}^{N} w_{1,2l-1}\mathbf{u}'_{2l-1,*} &= WD_3W^{\mathrm{T}}B^{\mathrm{T}}B_r(\mathbf{v}_{0,*} + \mathbf{v}_{2N+1,*}) - WD_4W^{\mathrm{T}}B^{\mathrm{T}}A_r(\mathbf{v}_{0,*} + \mathbf{v}_{2N+1,*}) \\ &\quad + \sum_{l=1}^{N}\left[w_{1,2l-1}v'_{2l-1,0}\mathbf{p}_1^{(2l-1)} + w_{1,2l-1}v'_{2l-1,2N+1}\mathbf{p}_2^{(2l-1)}\right], \end{aligned} \tag{4.45}$$

where

$$D_1 = \sum_{l=1}^{N} w_{1,2l}\Lambda_{2l}^{-1}\left[\delta_{2l}I_{2N} - \gamma_{2l}\Lambda_{2l}^{-1}\right], \quad D_2 = \sum_{l=1}^{N} w_{1,2l}\delta_{2l}\Lambda_{2l}^{-2},$$
$$D_3 = \sum_{l=1}^{N} w_{1,2l-1}\Lambda_{2l-1}^{-1}\left[\delta_{2l-1}I_{2N} - \gamma_{2l-1}\Lambda_{2l-1}^{-1}\right], \quad D_4 = \sum_{l=1}^{N} w_{1,2l-1}\delta_{2l-1}\Lambda_{2l-1}^{-2}, \tag{4.46}$$

and $\Lambda_k = \Lambda + \lambda_k I_{2N}$. For each $k = 1,\ldots,2N$, the vectors $\mathbf{p}_1^{(k)}$, $\mathbf{p}_2^{(k)}$ of (4.39) are precomputed at a cost $\mathrm{O}(N)$ using COLROW [7,8]. By Remark 2.2, the vectors $\mathbf{z}_1^{(k)}$, $\mathbf{z}_2^{(k)}$ of (4.41) are precomputed at a cost $\mathrm{O}(N)$ using precomputed $\tilde{R}^{(k)}$ of (4.34). The diagonal matrices D_1, D_2, D_3, D_4 of (4.46) are precomputed at a cost $\mathrm{O}(N)$. Hence $v'_{2l,0}$, $v'_{2l-1,0}$, $v'_{2l,2N+1}$, $v'_{2l-1,2N+1}$ of (4.40) are computed at a cost $16N^2$. It follows from Remarks 2.2 and 2.3 that the sums (4.44) and (4.45) are computed at a cost $16N^2$. Thus (4.42) and (4.43) imply that $\mathbf{u}_{1,*}$ and $\mathbf{u}_{2N,*}$ are computed at a cost $32N^2$. Using again Remark 2.2, we obtain the following remark.

Remark 4.4. The cost of multiplying a vector by $\hat{S}$ by is $32N^2$.

The remainder of this section is concerned with the selection of a preconditioner $\tilde{P}$ for $\hat{S}$ and the solution of a linear system with $\tilde{P}$. First we split vector $\mathbf{v}$ of (3.9) into three parts,

$$\mathbf{v}_r = [v_{1,1},\ldots,v_{1,2N},\ldots,v_{2N,1},\ldots,v_{2N,2N}]^{\mathrm{T}},$$

and

$$\mathbf{v}_{*,0} = [v_{1,0},\ldots,v_{2N,0}]^{\mathrm{T}}, \quad \mathbf{v}_{*,2N+1} = [v_{1,2N+1},\ldots,v_{2N,2N+1}]^{\mathrm{T}}.$$

(The vector $\mathbf{v}_r$ is obtained by deleting from $\mathbf{v}$ the components of $\mathbf{v}_{*,0}$ and $\mathbf{v}_{*,2N+1}$.) Then (4.17), with $\mathbf{g} = \mathbf{f} = \mathbf{0}$, $\mathbf{a} = B^{-1}B^{-\mathrm{T}}\mathbf{c}$, and $\mathbf{b} = B^{-1}B^{-\mathrm{T}}\mathbf{d}$ can be written in the compact form

$$P_{11}\begin{bmatrix}\mathbf{u}_e\\ \mathbf{v}_r\end{bmatrix} + P_{12}\begin{bmatrix}\mathbf{v}_{*,0}\\ \mathbf{v}_{*,2N+1}\end{bmatrix} = \mathbf{0},$$
$$P_{21}\begin{bmatrix}\mathbf{u}_e\\ \mathbf{v}_r\end{bmatrix} = \begin{bmatrix}B^{-1}B^{-\mathrm{T}}\mathbf{c}\\ B^{-1}B^{-\mathrm{T}}\mathbf{d}\end{bmatrix}, \tag{4.47}$$

where

$$P_{11} = \begin{bmatrix} A\otimes B + B\otimes A & B\otimes B\\ O & A\otimes B + B\otimes A\end{bmatrix}, \tag{4.48}$$

and P_{12}, P_{21} are the corresponding blocks in (4.17).

The matrix P_{11} of (4.48) is nonsingular since, with $\mathbf{u}_e$ of the form (4.18), the equation $(A\otimes B + B\otimes B)\mathbf{u}_e = \mathbf{0}$ is the matrix–vector form of the following OSC problem: find $U \in \mathscr{M}_h^0 \otimes \mathscr{M}_h^0$ such that

$$-\Delta U(\xi_i, \xi_j) = 0, \quad i,j = 1,\ldots,2N.$$

It is well known (see, for example, [16, Proposition 3.1]) that the only solution to this problem is $U = 0$ which gives $\mathbf{u}_e = \mathbf{0}$.

Using nonsingularity of P_{11}, we eliminate $\begin{bmatrix}\mathbf{u}_e\\ \mathbf{v}_r\end{bmatrix}$ from (4.47) to obtain

$$\hat{P}\begin{bmatrix}\mathbf{v}_{*,0}\\ \mathbf{v}_{*,2N+1}\end{bmatrix} = \begin{bmatrix}\mathbf{c}\\ \mathbf{d}\end{bmatrix}, \tag{4.49}$$

where $\hat{P} = (I_2 \otimes B^{\mathrm{T}}B)P$ and the $4N \times 4N$ Schur complement matrix $P = -P_{21}P_{11}^{-1}P_{12}$.

Theorem 4.2. *The matrix $\hat{P}$ is symmetric and positive definite.*

Proof. The proof of this theorem is similar to that of Theorem 4.1. First we show that $\hat{P}$ is nonsingular. Then we prove that $\hat{P} = \hat{P}^T$ and that $\hat{P}$ is nonnegative definite. This and nonsingularity of $\hat{P}$ imply that $\hat{P}$ is positive definite. □

Since (4.47) is the same as (4.17) with $\mathbf{g} = \mathbf{f} = \mathbf{0}$, $\mathbf{a} = B^{-1}B^{-T}\mathbf{c}$, $\mathbf{b} = B^{-1}B^{-T}\mathbf{d}$, and since (4.49) was obtained from (4.47), we have the following remark.

Remark 4.5. For arbitrary $\mathbf{c}$ and $\mathbf{d}$, the solution of (4.49) is obtained by solving (4.17), with $\mathbf{g} = \mathbf{f} = \mathbf{0}$, $\mathbf{a} = B^{-1}B^{-T}\mathbf{c}$, and $\mathbf{b} = B^{-1}B^{-T}\mathbf{d}$, for the components $\{v_{k,0}\}_{k=1}^{2N}$ and $\{v_{k,2N+1}\}_{k=1}^{2N}$ of $\mathbf{v}$.

Since the orders of $\hat{S}$ of (4.8) and $\hat{P}$ are not the same, as a preconditioner for $\hat{S}$ we take the $(4N-4) \times (4N-4)$ matrix $\tilde{P}$ which arises on the left-hand side of (4.49) from the elimination of $v_{1,0}$, $v_{2N,0}$, $v_{1,2N+1}$, $v_{2N,2N+1}$ using the submatrix of $\hat{P}$ formed by the entries of $\hat{P}$ located in rows and columns 1, $2N$, $2N+1$, $4N$. Clearly, $\tilde{P}$ is symmetric and positive definite since it is the Schur complement of a symmetric and positive definite submatrix in the symmetric and positive definite $\hat{P}$. Moreover, for arbitrary $\{c_k\}_{k=2}^{2N-1}$ and $\{d_k\}_{k=2}^{2N-1}$, the solution of

$$\tilde{P}[v_{2,0}, \dots, v_{2N-1,0}, v_{2,2N+1}, \dots, v_{2N-1,2N+1}]^T = [c_2, \dots, c_{2N-1}, d_2, \dots, d_{2N-1}]^T \tag{4.50}$$

is obtained by solving (4.49), with $\mathbf{c} = [0, c_2, \dots, c_{2N-1}, 0]^T$, $\mathbf{d} = [0, d_2, \dots, d_{2N-1}, 0]^T$, for $\{v_{k,0}\}_{k=2}^{2N-1}$ and $\{v_{k,2N+1}\}_{k=2}^{2N-1}$. By Remark 4.5, we find the solution of (4.50) by solving (4.17), with $\mathbf{g} = \mathbf{f} = \mathbf{0}$, $\mathbf{a} = B^{-1}B^{-T}\mathbf{c}$, and $\mathbf{b} = B^{-1}B^{-T}\mathbf{d}$, for the components $\{v_{k,0}\}_{k=2}^{2N-1}$ and $\{v_{k,2N+1}\}_{k=2}^{2N-1}$ of $\mathbf{v}$. By Remark 2.3, the cost of Step 1 of Algorithm II, which involves computing $\mathbf{a}'$ and $\mathbf{b}'$ only, is $O(N \log_2 N)$. The cost of Step 2 to compute $v'_{k,0}$, $v'_{k,2N+1}$, $k = 1, \dots, 2N$, using (4.32) with $\mathbf{g}'_{k,*} = \mathbf{f}'_{k,*} = \mathbf{0}$ and the precomputed matrices $R^{(k)}$ of (4.33), is $O(N)$. By Remark 2.3, the cost of Step 3, which involves two multiplications by W to obtain $\{v_{k,0}\}_{k=1}^{2N}$ and $\{v_{k,2N+1}\}_{k=1}^{2N}$, is $O(N \log_2 N)$. Thus we arrive at the following remark.

Remark 4.6. The cost of solving a linear system with the preconditioner $\tilde{P}$ is $O(N \log_2 N)$.

With the preconditioner $\tilde{P}$, the convergence rate of the PCG method applied to a linear system with $\hat{S}$ depends on

$$\kappa_2(\tilde{P}^{-1/2}\hat{S}\tilde{P}^{-1/2}) = \lambda_{\max}(\tilde{P}^{-1}\hat{S})/\lambda_{\min}(\tilde{P}^{-1}\hat{S}).$$

The right-hand side in this formula was used to compute $\kappa_2(\tilde{P}^{-1/2}\hat{S}\tilde{P}^{-1/2})$ numerically for several values of N. Based on the results presented in Table 1, we conjecture that $\kappa_2(\tilde{P}^{-1/2}\hat{S}\tilde{P}^{-1/2})$ is bounded from above by a small positive constant which is independent of N. In comparison, $\kappa_2(\hat{S})$ is large and depends linearly on N.

Table 1
$\kappa_2(\tilde{P}^{-1/2}\hat{S}\tilde{P}^{-1/2})$ and $\kappa_2(\hat{S})$

N	$\kappa_2(\tilde{P}^{-1/2}\hat{S}\tilde{P}^{-1/2})$	$\kappa_2(\hat{S})$
4	2.69	0.986 + 03
8	3.08	0.277 + 04
16	3.32	0.598 + 04
32	3.46	0.122 + 05
64	3.55	0.245 + 05
128	3.61	0.490 + 05

4.4. Final implementation and cost of Algorithm I

In this section we discuss the final implementation and give the cost of Algorithm I of Section 4.1 for solving (3.11)–(3.13).

Step 1 of Algorithm I involves computing $S_{21}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix}$, where $S_{11}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{f}\end{bmatrix}$. Remark 4.1 implies that only subvectors $\mathbf{p}_{1,*}$ and $\mathbf{p}_{2N,*}$ of $\mathbf{p}$ are needed when solving the linear system with S_{11}. It follows from Remark 4.2 and the first part of Remark 4.3 that the cost of computing $\mathbf{f}' = (W^{\mathrm{T}}B^{\mathrm{T}} \otimes I_{2N})\mathbf{f}$ is $10N^2\log_2 N$ ($\mathbf{g}' = \mathbf{0}$ and $\mathbf{a}' = \mathbf{b}' = \mathbf{0}$ by (4.23)). Next, by the second part of Remark 4.3, we compute at a cost $\mathrm{O}(N^2)$, and save, the solutions $\mathbf{p}'_{k,*}, \mathbf{q}'_{k,*}$, $k = 1, \ldots, 2N$, of (4.28) with $\mathbf{g}'_{k,*} = \mathbf{0}$ and $a'_k = b'_k = 0$. Then $\mathbf{p}_{1,*}$ and $\mathbf{p}_{2N,*}$ are computed at a cost $\mathrm{O}(N^2)$ using (4.42) and (4.43) with $\mathbf{u}'_{k,*}$ replaced by $\mathbf{p}'_{k,*}$. Thus the cost of Step 1 is $10N^2\log_2 N$.

By Remark 2.2 the cost of the first part of Step 2 is $\mathrm{O}(N)$. The second part of Step 2 is carried out using the PCG method with $\tilde{P}$ as a preconditioner for $\hat{S}$. It follows from Remarks 4.4 and 4.6 that the cost of each PCG iteration is $32N^2$. Hence, with the number of PCG iterations equal to m, the cost of Step 2 is $32mN^2$.

To carry out Step 3, we first examine $\begin{bmatrix}\mathbf{w}\\ \mathbf{z}\end{bmatrix} = S_{11}^{-1}\begin{bmatrix}\mathbf{g}\\ \mathbf{f}\end{bmatrix}$, where $\begin{bmatrix}\mathbf{g}\\ \mathbf{f}\end{bmatrix}$ is given by (4.35). It follows from (4.36), (4.37), and Remark 2.2 that the subvectors $\mathbf{g}'_{k,*}$, $\mathbf{f}'_{k,*}$, $k = 1, \ldots, 2N$, of $\mathbf{g}' = (W^{\mathrm{T}}B^{\mathrm{T}} \otimes I_{2N})\mathbf{g}$, $\mathbf{f}' = (W^{\mathrm{T}}B^{\mathrm{T}} \otimes I_{2N})\mathbf{f}$ are computed at a cost $\mathrm{O}(N^2)$. Then, by the second part of Remark 4.3, the solutions $\mathbf{w}'_{k,*}, \mathbf{z}'_{k,*}$, $k = 1, \ldots, 2N$, of (4.28), with $a'_k = b'_k = 0$, are computed at a cost $\mathrm{O}(N^2)$. Finally, to obtain $\mathbf{u}$ of

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} = S_{11}^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{f}\end{bmatrix} - S_{11}^{-1}\begin{bmatrix}\mathbf{g}\\ \mathbf{f}\end{bmatrix},$$

we first compute $\mathbf{u}'_{k,*} = \mathbf{p}'_{k,*} - \mathbf{w}'_{k,*}$, $k = 1, \ldots, 2N$, at a cost $\mathrm{O}(N^2)$, and then, by the third part of Remark 4.3, we compute $\mathbf{u}_e = (W \otimes I_{2N})\mathbf{u}'$ (and hence its restriction $\mathbf{u}$) at a cost $10N^2\log_2 N$. If $\mathbf{v}$ is also required, then $\mathbf{v}'_{k,*} = \mathbf{q}'_{k,*} - \mathbf{z}'_{k,*}$, $k = 1, \ldots, 2N$, and $\mathbf{v} = (W \otimes I_{2N+2})\mathbf{v}'$ are computed at costs $\mathrm{O}(N^2)$ and $10N^2\log_2 N$, respectively.

It follows from the above discussion that the cost of solving (3.11)–(3.13) for $\mathbf{u}$ is

$$20N^2\log_2 N + 32mN^2, \tag{4.51}$$

where m is the number of PCG iterations.

5. Numerical results

We used the method of this paper to solve several test problems corresponding to (1.1). The algorithm was run in double precision on a Gateway PC E-2000 400. For an $N \times N$ uniform partition of Ω, the initial guess for the PCG method was $\mathbf{0}$ and the number m of PCG iterations was set to $\log_2 N + 2$. By (4.51), the cost of computing the $4N^2$ coefficients in the OSC approximation to u was $52N^2\log_2 N$.

Convergence rates in the maximum norm $\|w\|_C = \max_{0\leqslant n,m\leqslant 200}|w(t_n, t_m)|$ and the nodal norm $\|w\|_{C_h} = \max_{0\leqslant n,m\leqslant N}|w(t_n, t_m)|$ were determined using the formula

$$\text{rate} = \frac{\log(e_{N/2}/e_N)}{\log 2},$$

where e_N is the error corresponding to the $N \times N$ partition.

Example 1. In this artificial example, which is the same as Test Problem 2 in [1], we take $\Omega = (0,1) \times (0,1)$ and f, $g_1 = g_2 \equiv 0$ corresponding to the smooth exact solution

$$u(x,y) = [1 - \cos(2\pi x)][1 - \cos(2\pi y)].$$

(This example is similar to that in [4,11], where $u(x,y) = \sin^2(2\pi x)\sin^2(2\pi y)$.) In [1], the corresponding finite difference linear system was solved using the full multigrid method and the iterative multigrid method in which W cycling was continued until the maximum correction by a single cycle was less than 10^{-11}. For the same values of $N = 16, 32, 64, 128$ as in [1], the values of our term $\log_2 N$ are smaller than the number of W cycles in [1, Table 2]. In [1], the cost of each W cycle is cN^2, where the constant c, not given in [1], is expected to be larger than our constant of 52. Therefore, on this test problem, our algorithm is competitive with the iterative W-cycle multigrid method of [1]. In Tables 2 and 3, we give errors and the corresponding convergence rates for u and v using the maximum and nodal norms. The convergence rates of order four for the approximations to the first order derivatives of u and v at the partition nodes demonstrate superconvergence phenomenon of OSC. The errors $\|u - U\|_{C_h}$ in Table 2 are comparable to those in column 4 of Table 2 in [1]. In comparison to [1], our method also produces approximations to $v = \Delta u$, v_x, and v_y. More importantly, our approximations to u, v and their first order derivatives are continuous piecewise polynomials defined over $\overline{\Omega}$.

Example 2. In this example, corresponding to bending of a square clamped plate under a uniform load [18], we take $\Omega = (0,1) \times (0,1)$, $f \equiv 1$, and $g_1 = g_2 \equiv 0$. Although the exact solution u is not known in closed form, it follows from [2, Remark 2.1] that u is, at least, in $H^{4.7}(\Omega)$. In Table 4, we present computed values

Table 2
Maximum and nodal errors, and convergence rates for u

N	$\|u-U\|_C$		$\|u-U\|_{C_h}$		$\|(u-U)_x\|_{C_h}$		$\|(u-U)_y\|_{C_h}$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
16	0.164 – 03		0.785 – 04		0.415 – 04		0.415 – 04	
32	0.105 – 04	3.965	0.486 – 05	4.015	0.248 – 05	4.063	0.248 – 05	4.063
64	0.605 – 06	4.118	0.303 – 06	4.004	0.153 – 06	4.016	0.153 – 06	4.016
128	0.397 – 07	3.930	0.189 – 07	4.001	0.955 – 08	4.004	0.955 – 08	4.004

Table 3
Maximum and nodal errors, and convergence rates for v

N	$\|v-V\|_C$		$\|v-V\|_{C_h}$		$\|(v-V)_x\|_{C_h}$		$\|(v-V)_y\|_{C_h}$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
16	0.929 – 02		0.509 – 02		0.122 – 01		0.122 – 01	
32	0.593 – 03	3.970	0.317 – 03	4.004	0.768 – 03	3.990	0.768 – 03	3.990
64	0.341 – 04	4.121	0.198 – 04	4.001	0.480 – 04	3.999	0.480 – 04	3.999
128	0.224 – 05	3.930	0.124 – 05	4.000	0.302 – 05	3.990	0.300 – 05	4.000

Table 4
Computed deflection and bending moments

N	$u(0.5, 0.5)$	$M_x(1, 0.5)$	$M_y(0.5, 1)$
4	0.00125862025	−0.0513059279	−0.0515000295
8	0.00126485585	−0.0513338674	−0.0513348569
16	0.00126528707	−0.0513335820	−0.0513335820
32	0.00126531700	−0.0513337482	−0.0513337482
64	0.00126531896	−0.0513337636	−0.0513337636
128	0.00126531908	−0.0513337647	−0.0513337647

of the deflection $u(0.5,0.5)$ and the bending moments $M_x(1,0.5)$, $M_y(0.5,1)$, where $M_x = -\Delta u$ and $M_y = -\Delta u$ on the vertical and horizontal sides of $\partial\Omega$, respectively. In comparison, the corresponding values reported in [18, Table 35] are $u(0.5,0.5) = 0.00126$, $M_x(1,0.5) = -0.0513$, and $M_y(0.5,1) = -0.0513$. These values were obtained in [18] by a Fourier series method and by replacing an infinite number of the resulting linear equations with four equations. With $N = 4$, we obtain three correct significant digits in our approximation to $u(0.5,0.5)$.

The same problem was solved in [11] using the mixed approach and the spectral collocation discretization with the Chebyshev polynomials of degree $\leqslant N$. The resulting linear system was solved by the Richardson method with a finite difference preconditioner. The errors reported in [11, Table VII] are comparable to our errors for $N = 8$ and are smaller for $N = 16, 32$ by the factor of 0.5×10^{-1}. It is noted in [11] that the spectral method does not converge exponentially due to the singularities at the corners.

Example 3. To show the applicability of our approach to the solution of other plate bending problems, in this example, we solve the problem

$$\Delta^2 u = 1 \text{ in } \Omega = (0,1)\times(0,1), \quad u = 0 \text{ on } \partial\Omega, \quad u_y(x,1) = 0,\ 0 < x < 1,$$

$$\Delta u(0,y) = \Delta u(1,y) = 0,\ 0 < y < 1, \quad \Delta u(x,0) = 0,\ 0 < x < 1.$$

This problem models bending of a square plate under a uniform load with one side clamped and the three remaining sides simply supported [18]. The exact solution $u \notin C^4(\overline{\Omega})$ since otherwise the boundary conditions and the differential equation would imply $\Delta^2 u(0,1) = 0$ and $\Delta^2 u(0,1) = 1$, respectively. Finding the OSC solution for this problem involves solving linear system (4.17) with $\mathbf{g} = \mathbf{0}$, $\mathbf{b} = \mathbf{0}$, and the condition $-\mathbf{u}_{*,1} = \mathbf{a}$ replaced by $\mathbf{v}_{*,0} = \mathbf{0}$. Such a system can be solved directly and the cost of computing $4N^2$ unknowns corresponding to the OSC approximation of u is $20N^2\log_2 N$. In Table 5 we present computed values of the deflection $u(0.5,0.5)$ and the bending moment $M_y(0.5,1)$, where $M_y = -\Delta u$ on the horizontal sides of $\partial\Omega$. In comparison, the corresponding values reported in [18, Table 32] are $u(0.5,0.5) = 0.0028$ and $M_y(0.5,1) = -0.084$.

Example 4. In this example, corresponding to bending of a square clamped plate under the load 1 concentrated at the center [18], we take $\Omega = (0,1)\times(0,1)$, $g_1 = g_2 \equiv 0$, and

$$f(x,y) = \begin{cases} 1/(4h^2) & \text{if } |x-1/2| \leqslant h \text{ and } |y-1/2| \leqslant h, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

In Table 6, we present computed values of the deflection $u(0.5,0.5)$ and the bending moment $M_y(0.5,1)$, where $M_y = -\Delta u$ on the horizontal sides of $\partial\Omega$. In comparison, the corresponding values reported in [18, Table 37] are $u(0.5,0.5) = 0.00560$ and $M_y(0.5,1) = -0.1257$. We obtain better approximations for the larger values of N since then $f(x,y)$ of (5.1) better models the load 1 concentrated at the center.

Table 5
Computed deflection and bending moment

N	$u(0.5,0.5)$	$M_y(0.5,1)$
4	0.00278243314	−0.0838262426
8	0.00278527850	−0.0838704717
16	0.00278548018	−0.0838748929
32	0.00278549313	−0.0838751788
64	0.00278549394	−0.0838751968
128	0.00278549399	−0.0838751979

Table 6
Computed deflection and bending moment

N	$u(0.5, 0.5)$	$M_y(0.5, 1)$
4	0.00338671561	−0.105221752
8	0.00476831786	−0.120689230
16	0.00532930384	−0.124502492
32	0.00552339288	−0.125453479
64	0.00558537771	−0.125691269
128	0.00560424025	−0.125750723

Example 5. In this example, corresponding to the problem (1)–(2) of [15] with Dw replaced by u, $p = p_0 \equiv 1$, $a = 2$, $b = 1$, we take $\Omega = (-2, 2) \times (-1, 1)$, $f \equiv 1$, and $g_1 = g_2 \equiv 0$. For $N = 128$, our plots in Figs. 1 and 2 reproduce very well the corresponding graphs of v_x and v_y ($v = \Delta u$) in Figures of 4 and 5 of [15].

Example 6. In this example, corresponding to creeping flow of a viscous incompressible fluid in a square cavity [12], we take $\Omega = (0, 1) \times (0, 1)$, $f \equiv 0$, $g_1 \equiv 0$, $g_2 = 1$ on the upper side of $\partial\Omega$, and $g_2 = 0$ on the remaining sides of $\partial\Omega$. The boundary conditions imply that u_y is discontinuous at the vertices $(0, 1)$ and $(1, 1)$ and it is well-known (see, e.g., [12]) that the vorticity $v = \Delta u$ is unbounded near the points $(0, 1)$ and

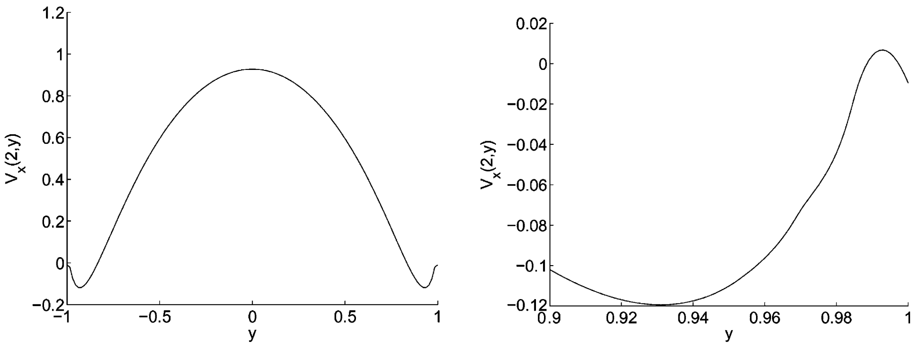

Fig. 1. $V_x(2, y)$ for y in $[-1, 1]$ and $[0.9, 1]$.

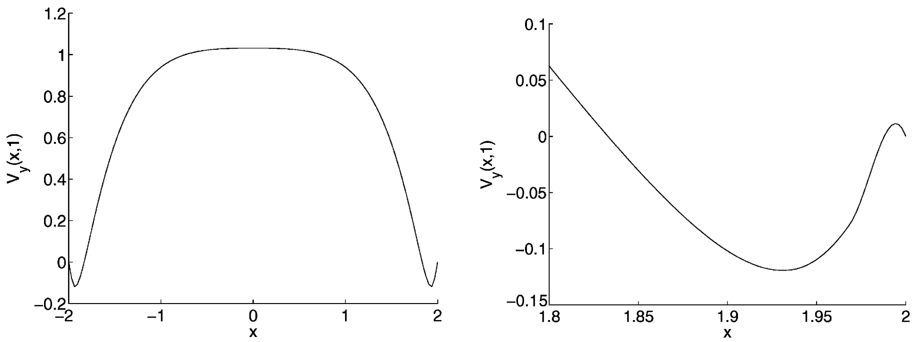

Fig. 2. $V_y(x, 1)$ for x in $[-2, 2]$ and $[1.8, 2]$.

Table 7
Primary vortex, stream function and vorticity

N	Vortex	u	v
4	(0.5,0.7534)	0.0981098667	−3.09335605
8	(0.5,0.7651)	0.100019733	−3.21563572
16	(0.5,0.765)	0.100100124	−3.21252713
32	(0.5,0.765)	0.100081897	−3.21195846
64	(0.5,0.765)	0.100076897	−3.21191707
128	(0.5,0.765)	0.100076276	−3.21192238

$(1,1)$. We approximate the nonzero boundary condition using the approach described in [13, Section 5.1]. Taking advantage of $f \equiv 0$, we reduce the cost of our algorithm to $42N^2 \log_2 N$ when computing $4N^2$ coefficients in the OSC approximation to u. The computed coordinates of the primary vortex center, the computed values of the stream function u and vorticity v at the vortex are reported in Table 7. They are in good agreement with the corresponding results of Kelmanson [12] who subtracted a singularity in his integral equation method to obtain $u = 0.0998$, $v = -3.2021$ at the vortex $(0.5, 0.76)$.

This problem was solved in [1] as Test Problem 3. Our coordinates of the vortex and our value of u at the vortex are very close to the corresponding results reported in [1, Table 3], where the iterative multigrid method with 14 W cycles was used for an $N \times N$ grid with $N = 16, 32, 64$. Since the cost of each W cycle is expected to be larger than $42N^2$ and since our term $\log_2 N$ is smaller than 14, it appears, that our algorithm is more efficient than that of [1] for solving this test problem. Moreover, in comparison to [1], we also obtain, if desired, an approximation to vorticity at the vortex.

The same problem was also solved in [20] using a spectral multigrid method. Although the reported results $(0.5, 0.78)$ for the vortex, $u = 0.09975$, and $v = -3.36$ are in reasonable agreement with those of [12], they are no more accurate than our results. This is expected since the solution u is not smooth.

6. Conclusions

In this paper we developed an efficient algorithm for solving the biharmonic Dirichlet problem on a rectangle using a fourth order mixed method based on the piecewise Hermite bicubic OSC discretization. The nearly optimal algorithm is faster than previously proposed algorithms for solving the same OSC linear system. The algorithm is competitive with other fourth order finite difference and finite element Galerkin algorithms and is particularly well suited for solving plate bending problems with different kinds of clamped and simply supported boundary conditions.

Acknowledgements

This work was supported in part by National Science Foundation grant DMS-9805827. The author wishes to thank Rakhim Aitbayev and Graeme Fairweather for their assistance during the preparation of this paper.

References

[1] I. Altas, J. Dym, M.M. Gupta, R.P. Manohar, Multigrid solution of automatically generated high-order discretizations for the biharmonic equation, SIAM J. Sci. Comput. 19 (1998) 1575–1585.

[2] C. Bernardi, Y. Maday, Spectral Methods, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, vol. V, Elsevier Science, 1997.

[3] B. Bialecki, G. Fairweather, K.R. Bennett, Fast direct solvers for piecewise Hermite bicubic orthogonal spline collocation equations, SIAM J. Numer. Anal. 29 (1992) 156–173.

[4] B. Bialecki, A. Karageorghis, A Legendre spectral collocation method for the biharmonic Dirichlet problem, Math. Mod. Numer. Anal. 34 (2000) 637–662.

[5] C. de Boor, A Practical Guide to Splines, Applied Mathematical Sciences, vol. 27, Springer, New York, 1978.

[6] P.G. Ciarlet, P.A. Raviart, A mixed finite element method for the biharmonic equation, in: C. de Boor (Ed.), Symposium on Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, 1974, pp. 125–143.

[7] J.C. Diaz, G. Fairweather, 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) 358–375.

[8] J.C. Diaz, G. Fairweather, 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) 376–380.

[9] J. Douglas Jr., T. Dupont, Collocation Methods for Parabolic Equations in a Single Space Variable, Lecture Notes in Mathematics, vol. 358, Springer, New York, 1974.

[10] M.R. Hanisch, Multigrid preconditioning for the biharmonic Dirichlet problem, SIAM J. Numer. Anal. 30 (1993) 184–214.

[11] W. Heinrichs, A spectral multigrid method for the Stokes problem in streamfunction formulation, J. Comput. Phys. 102 (1992) 310–318.

[12] M.A. Kelmanson, Modified integral equation solution of viscous flows near sharp corners, Comput. Fluids 11 (1983) 307–324.

[13] Z.-M. Lou, B. Bialecki, G. Fairweather, Orthogonal spline collocation methods for biharmonic problems, Numer. Math. 80 (1998) 267–303.

[14] V.V. Meleshko, Biharmonic problem in a rectangle, Appl. Scientific Res. 58 (1998) 217–249.

[15] V.V. Meleshko, A.M. Gomilko, A.A. Gourjii, Normal reactions in a clamped elastic rectangle plate, J. Eng. Math. 40 (2001) 377–398.

[16] P. Percell, M.F. Wheeler, A C^1 finite element collocation method for elliptic equations, SIAM J. Numer. Anal. 17 (1980) 605–622.

[17] W. Sun, Orthogonal collocation solution of biharmonic equations, Int. J. Comput. Math. 49 (1993) 221–232.

[18] S. Timoshenko, S. Woinowsky-Krieger, Theory of Plates and Shells, McGraw-Hill, New York, 1959.

[19] C. Van Loan, Computational Frameworks for the Fast Fourier Transform, SIAM, Philadelphia, 1992.

[20] R.W. DeVries, P.J. Zandbergen, The numerical solution of the biharmonic equation, using a spectral multigrid method, in: W.F. Ballhaus, M.Y. Hussaini (Eds.), Advances in Fluid Dynamics, Springer, New York, 1989, pp. 25–35.

[21] X. Zhang, Multilevel Schwarz methods for the biharmonic Dirichlet problem, SIAM. J. Sci. Comput. 15 (1994) 621–644.