Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Computational Physics 191 (2003) 476–501

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Domain decomposition algorithms for fourth-order nonlinear elliptic eigenvalue problems ☆

S.-L. Chang [a], C.-S. Chien [b,*]

[a] Center for General Education, Southern Taiwan University of Technology, Tainan 710, Taiwan, ROC
[b] Department of Applied Mathematics, National Chung-Hsing University, Taichung 402, Taiwan, ROC

Received 16 November 2001; received in revised form 2 June 2003; accepted 20 June 2003

Abstract

We study domain decomposition methods for fourth-order plate problems. The well-known von Kármán equations are used as our model problem. By exploiting the symmetry of the domain, the solution of the original problem can be obtained by solving those associated reduced problems, which are defined on subdomains with appropriate boundary conditions. We show how nonoverlapping and overlapping domain decomposition methods can be used to solve the reduced problems. For the linearized von Kármán equation, we present preconditioners using both Fourier analysis and probing techniques for the interface systems, which are similar to those derived by Chan et al. Finally, we compare the efficiency of various domain decomposition preconditioners for solving the von Kármán equations.

AMS: 65N55; 65N06; 73H05; 65F10

Keywords: Domain decomposition; Preconditioning; Symmetry; Plate problems; Partially clamped boundary conditions

1. Introduction

Here, we consider fourth-order nonlinear elliptic eigenvalue problems of the following form:

$$G(u,\lambda) = 0, \tag{1.1}$$

where $G : X \times \mathbb{R}^k \to X$ is a smooth operator, X is some Banach space, $u \in X$ and $\lambda \in \mathbb{R}^k, k \geqslant 1$. Here, u represents a solution field (e.g., displacements) and λ is a real vector of physical parameters. We will solve Eq. (1.1) numerically by the continuation method, based on parametrizing the solution branch by arc-length, say $[u(s), \lambda(s)]$. First we discretize Eq. (1.1), for example, by a finite difference method or a finite

☆ Supported by the National Science Council of ROC (Taiwan) through Project NSC 90-2115-M-005-007.
* Corresponding author. Fax: +886-4-287-3028.
E-mail address: cschien@amath.nchu.edu.tw (C.-S. Chien).

0021-9991/$ - see front matter
doi:10.1016/S0021-9991(03)00327-9

element method. In both cases, Eq. (1.1) is approximated by a finite-dimensional problem of the following form:

$$H(x, \lambda) = 0, \tag{1.2}$$

where $H : \mathbb{R}^N \times \mathbb{R}^k \to \mathbb{R}^N$ is a smooth mapping of $x \in \mathbb{R}^N$ and $\lambda \in \mathbb{R}^k$. Viewing some component of λ as the continuation parameter, the continuation algorithms can be implemented to trace solution curves of Eq. (1.2).

A typical example of Eq. (1.1) is the well-known von Kármán equations [11, pp. 439–453]. Recently Chien and coworkers [6,7] investigated numerical solutions of the von Kármán equations, where some conjugate gradient type methods were incorporated in the context of continuation methods to trace solution curves and detect bifurcation points. In particular, Chien et al. [8] studied symmetry and scaling properties of the von Kármán equations

$$\Delta^2 w + \lambda \frac{\partial^2 w}{\partial x^2} - [f, w] = 0, \qquad \Delta^2 f + \tfrac{1}{2}[w, w] = 0 \quad \text{in } \Omega = [0, l] \times [0, 1] \tag{1.3}$$

with simply supported boundary conditions

$$w = \Delta w = 0, \qquad f = \Delta f = 0 \quad \text{on } \partial\Omega. \tag{1.4}$$

Here Ω represents the shape of a rectangular plate in its flat state, $f(x, y)$ is the Airy stress function describing the averaged stress over the thickness of the plate, $w(x, y)$ is the deformation of the plate under the action of the external load λ, and the bracket operator $[\cdot, \cdot]$ is defined by

$$[u, v] = u_{xx}v_{yy} - 2u_{xy}v_{xy} + u_{yy}v_{xx}.$$

By exploiting symmetries of the domain, Chien et al. showed that solution branches of (1.3) with boundary conditions (1.4) can be represented by solving the associated reduced problems defined on the fundamental domains $\mathscr{C} = [0, l/2m] \times [0, 1/2n], m, n \in \mathbb{N}$. Here, both w and f satisfy the boundary conditions

$$\begin{aligned} &u = \Delta u = 0 \quad \text{on } x = 0 \text{ and } y = 0, \\ &\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial^3 u}{\partial \mathbf{n}^3} = 0 \quad \text{on } x = \frac{l}{2m} \text{ and } y = \frac{1}{2n}. \end{aligned} \tag{1.5}$$

In this paper, we will show how the domain decomposition methods can be used to solve fourth-order nonlinear eigenvalue problems. In the past two decades, domain decomposition has been a very popular research topic, partly because of the potential of parallel implementation [13,15]. For second-order elliptic problems, there are plenty of published research articles, see e.g., [2] and the further references cited therein. One of the main issues concerning domain decomposition methods is how to design efficient preconditioners for the interface operators corresponding to the Laplace equation defined on a rectangle which is decomposed into two subrectangles, see e.g., [15]. For fourth-order problems, less satisfactory results are known. In [5] Chan et al. presented preconditioners for the interface system arising from solving the following biharmonic equation with 13-point finite difference discretizations

$$\Delta^2 w = q \quad \text{in } \Omega = [0, 1]^2. \tag{1.6}$$

Here, three kinds of boundary conditions are imposed on $\partial\Omega$:

(i) The simply supported boundary conditions

$$w = \Delta w = 0 \quad \text{on } \partial\Omega. \tag{1.7}$$

(ii) The clamped boundary conditions

$$w = w_{\mathbf{n}} = 0 \quad \text{on } \partial\Omega. \tag{1.8}$$

(iii) The mixed boundary conditions

$$w|_{\partial\Omega} = \Delta w|_{\partial\Omega_x} = w_{\mathbf{n}}|_{\partial\Omega_y} = 0, \tag{1.9}$$

where $\partial\Omega_x$ and $\partial\Omega_y$ denote the part of $\partial\Omega$ that are parallel to the x- and y-axis, respectively. By using discrete Fourier analysis, they obtained the exact eigen-decomposition of the interface Schur complement for Eq. (1.6) with boundary conditions (1.7) and (1.8), respectively.

Our aim here is to design efficient preconditioners when the 13-point centered difference approximations combined with domain decomposition method are exploited to solve Eq. (1.1), where both of the nonoverlapping and overlapping domain decomposition are considered. In the discussion given below, we use the von Kármán equations as our example. However, our results can be applied to any fourth-order problems as well.

This paper is organized as follows: In Section 2, we briefly review a domain decomposition method for the linearized von Kármán equation, where only nonoverlapping domain decomposition is considered. We obtain results similar to those of Chan et al. [5]. In Section 3, we discuss how nonoverlapping and overlapping domain decomposition can be incorporated in the context of continuation methods to solve fourth-order nonlinear elliptic eigenvalue problems. In particular, the probing technique discussed in Section 2 is used as a preconditioner for overlapping domain decomposition. Our numerical results are reported in Section 4. Finally, some concluding remarks are given in Section 5.

2. A brief review of domain decomposition method for linear problems

For simplicity, we rewrite the linearized von Kármán equation with simply supported boundary conditions as follows:

$$\begin{aligned} &\Delta^2 w + \lambda w_{xx} = 0 \quad \text{in } \Omega^* = [0,1] \times [0,1], \\ &w = \Delta w = 0 \quad \text{on } \partial\Omega^*. \end{aligned} \tag{2.1}$$

The eigenpairs of (2.1) are

$$\begin{aligned} &\lambda_{m,n} = \left(\frac{\pi}{m}\right)^2 (m^2 + n^2)^2, \\ &w_{m,n}(x,y) = \sin m\pi x \sin n\pi y, \quad m,n = 1,2,3,\ldots \end{aligned} \tag{2.2}$$

By exploiting the rule of separation of variables, the first eigenpair of (2.1) can be obtained by solving the following associated reduced problem:

$$\begin{aligned} &\Delta^2 w + \lambda w_{xx} = 0 \quad \text{in } \Omega = [0,\tfrac{1}{2}] \times [0,\tfrac{1}{2}], \\ &w = \Delta w = 0 \quad \text{on } x = 0 \text{ and } y = 0, \\ &w_{\mathbf{n}} = w_{\mathbf{nnn}} = 0 \quad \text{on } x = \tfrac{1}{2} \text{ and } y = \tfrac{1}{2}. \end{aligned} \tag{2.3}$$

The eigenpairs of (2.3) are

$$\begin{aligned} &\lambda_{m,n} = \begin{cases} 2\pi^2(2n+1)^2 \pm \pi^2\sqrt{2((2n+1)^4 + (2m+1)^4)}, & m < n, \\ 4\pi^2(2n+1)^2, & m = n, \end{cases} \\ &w_{m,n}(x,y) = \sin(2m+1)\pi x \sin(2n+1)\pi y, \quad m,n = 0,1,2,\ldots \end{aligned} \tag{2.4}$$

Setting $m = n = 1$ in (2.2) and $m = n = 0$ in (2.4), it is clear that the first eigenpair of (2.1) and (2.3) are the same.

Suppose that Ω is decomposed into two subdomains Ω_1 and Ω_2 with interface Γ, as shown in Fig. 1. We also suppose that a uniform mesh with size h is used on Ω, with L grid points in the y-axis. The simplest way of decoupling the two subdomains is to introduce two computational grid interfaces Ω_3^h and Ω_4^h near the physical interface Γ. Assume that

$$l_1 = m_1 h, \quad l_2 = (m_2 - 1)h,$$

where m_1 and m_2 denote the number of grid points along the y-axis of the two subdomains Ω_1^h and Ω_2^h. Note that $L = m_1 + m_2 + 2$. Suppose that the unknowns are ordered in such a way that the interior points in the subdomains appear first and those on the two interfaces appear last, then the discrete solution vector w can be expressed as $w = (w_1, w_2, w_3, w_4)$, where the w_is denote the unknowns on Ω_i^h.

Let A and $D \in \mathbb{R}^{L^2 \times L^2}$ be the discretization matrices corresponding to the differential operators Δ^2 and $\partial^2/\partial x^2$, respectively. We have

$$A = \frac{1}{h^4}\left[\begin{array}{ccccccc|ccccccc|c|c}
A_1 & B & I & & & & & & & & & & & & B & I \\
B & A_1 & B & I & & & & & & & & & & & I & \\
I & B & A_1 & B & I & & & & & & & & & & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & & & & & & \\
 & & \ddots & \ddots & \ddots & \ddots & I & & & & & & & & & \\
 & & & I & B & A_1 & B & & & & & & & & & \\
 & & & & I & B & A_2 & & & & & & & & & \\
\hline
 & & & & & & & A_1 & B & I & & & & & I & B \\
 & & & & & & & B & A_1 & B & I & & & & & I \\
 & & & & & & & I & B & A_1 & B & I & & & & \\
 & & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
 & & & & & & & & & \ddots & \ddots & \ddots & \ddots & I & & \\
 & & & & & & & & & & I & B & A_3 & B & & \\
 & & & & & & & & & & & 2I & 2B & A_1 & & \\
\hline
B & I & & & & & & I & & & & & & & A_1 & B \\
\hline
I & & & & & & & B & I & & & & & & B & A_1
\end{array}\right] \in \mathbb{R}^{L^2 \times L^2} \qquad (2.5)$$

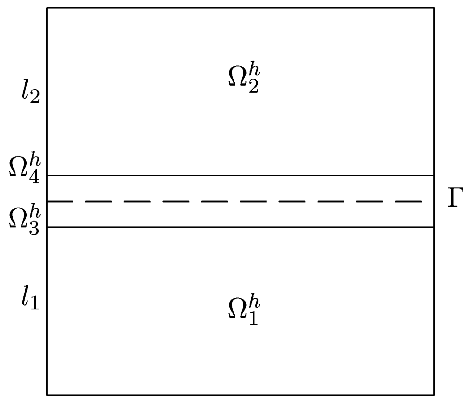

Fig. 1. The domain Ω and its partition.

with

$$B = \begin{bmatrix} -8 & 2 & & & & \mathbf{0} \\ 2 & -8 & 2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & 2 & \\ & & & 2 & -8 & 2 \\ \mathbf{0} & & & & 4 & -8 \end{bmatrix} \in \mathbb{R}^{L\times L},$$

$$A_1 = \begin{bmatrix} 19 & -8 & 1 & & & & \mathbf{0} \\ -8 & 20 & -8 & 1 & & & \\ 1 & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 20 & -8 & 1 \\ & & & 1 & -8 & 21 & -8 \\ \mathbf{0} & & & & 2 & -16 & 20 \end{bmatrix} \in \mathbb{R}^{L\times L},$$

$A_2 = A_1 - I$, $A_3 = A_1 + I$, I is the $L \times L$ identity matrix, and

$$D = \operatorname{diag}(D_L, \ldots, D_L) \tag{2.6}$$

with

$$D_L = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \mathbf{0} \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ \mathbf{0} & & & 2 & -2 \end{bmatrix} \in \mathbb{R}^{L\times L}.$$

Hence, the discretization system corresponding to Eq. (2.3) is

$$H(w, \lambda) = Aw + \lambda Dw = 0, \tag{2.7}$$

where $H : \mathbb{R}^{L^2} \times \mathbb{R} \to \mathbb{R}^{L^2}$ is a smooth mapping of $w \in \mathbb{R}^{L^2}$ and $\lambda \in \mathbb{R}$. We denote the Jacobian of H by $DH = [D_w H, D_\lambda H]$ and the solution curve c of (2.7) by

$$c = \{y(s) = (w(s), \lambda(s)) \,|\, H(y(s)) = 0, s \in I \subset \mathbb{R}\}.$$

In predictor–corrector continuation methods, we need to solve linear systems of the following form:

$$\begin{bmatrix} \bar{A} & p \\ q^{\mathrm{T}} & \gamma \end{bmatrix}\begin{bmatrix} w \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{2.8}$$

where $p, q, f \in \mathbb{R}^{L^2}$ and $\gamma, g \in \mathbb{R}$. The block elimination algorithm [12] is given as follows:

Algorithm 2.1 (*Block elimination*).
Step 1. Solve $\bar{A}u = p$, $\bar{A}v = f$.
Step 2. Compute $\lambda = (g - q^{\mathrm{T}}v)/(\gamma - q^{\mathrm{T}}u)$.
Step 3. Compute $w = v - \lambda u$.

2.1. Fourier analysis of the interface operator

Now we discuss preconditioners for the interface systems arising from solving fourth-order plate problems. For simplicity, we rewrite the linear systems in Step 1 of Algorithm 2.1 as

$$\bar{A}w = b,$$

where $\bar{A} = D_w H(y_i) = A(y_i) + \lambda D(y_i)$, which can be expressed in block form as:

$$\begin{bmatrix} A_{11} & 0 & A_{13} & A_{14} \\ 0 & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{23}^{\mathrm{T}} & A_{33} & A_{34} \\ A_{14}^{\mathrm{T}} & A_{42} & A_{34}^{\mathrm{T}} & A_{44} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}. \tag{2.9}$$

From the linear system (2.9) we obtain

$$w_1 = A_{11}^{-1}(b_1 - A_{13}w_3 - A_{14}w_4)$$

and

$$w_2 = A_{22}^{-1}(b_2 - A_{23}w_3 - A_{24}w_4).$$

On substituting w_1 and w_2 into (2.9), we obtain the following reduced system:

$$C\begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} g_3 \\ g_4 \end{bmatrix}, \tag{2.10}$$

where

$$C = \begin{bmatrix} A_{33} & A_{34} \\ A_{34}^{\mathrm{T}} & A_{44} \end{bmatrix} - \begin{bmatrix} A_{31} & A_{23}^{\mathrm{T}} \\ A_{14}^{\mathrm{T}} & A_{42} \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & A_{22}^{-1} \end{bmatrix} \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix} \equiv \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

is the Schur complement matrix corresponding to the reduced interface operator, and

$$g_3 = b_3 - A_{31}A_{11}^{-1}b_1 - A_{23}^{\mathrm{T}}A_{22}^{-1}b_2,$$

$$g_4 = b_4 - A_{14}^{\mathrm{T}}A_{11}^{-1}b_1 - A_{42}A_{22}^{-1}b_2.$$

The blocks C_{11} and C_{22} account for the coupling of the unknowns on Ω_1^h and Ω_2^h, respectively, among themselves, and the blocks C_{12} and C_{21} account for the coupling between the unknowns on the two interfaces.

Defining $b_1' = A_{11}^{-1}b_1$ and $b_2' = A_{22}^{-1}b_2$, the solution to the linear system (2.9) is obtained by applying the block elimination once more.

Algorithm 2.2 (*Block elimination algorithm for solving* (2.9)).

Step 1. Solve $A_{11}b_1' = b_1, A_{22}b_2' = b_2$, successively.

Step 2. Compute $g_3 = b_3 - A_{31}b_1' - A_{23}^{\mathrm{T}}b_2'$, $g_4 = b_4 - A_{14}^{\mathrm{T}}b_1' - A_{42}b_2'$.

Step 3. Solve

$$C\begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} g_3 \\ g_4 \end{bmatrix}.$$

Step 4. Compute $g_1 = b_1 - A_{13}w_3 - A_{14}w_4$, $g_2 = b_2 - A_{23}w_3 - A_{24}w_4$.

Step 5. Solve $A_{11}w_1 = g_1$, $A_{22}w_2 = g_2$, respectively, or on a parallel computer.

Since C is dense, unsymmetric and expensive to form explicitly, the preconditioned iterative methods are usually preferred to solve the linear system (2.10). In this paper, the preconditioned GMRES is implemented. The key point is to find an efficient preconditioner for C. In an approach similar to that described in [5], we use discrete Fourier analysis for eigen-decomposition of the Schur complement matrix C. For the problem (2.3), however, the computations are more complicated than those given in [5].

Denote by $v_j,\ j = 1, \ldots, L$, the eigenvectors of the one-dimensional discrete Laplace operator:

$$v_j = \sqrt{4h}[\sin(2j\pi h), \sin(4j\pi h), \ldots, \sin(2Lj\pi h)]^{\mathrm{T}},$$

where $h = 1/2(L+1)$. Next, let $\mathscr{V} = [v_1, v_2, \ldots, v_L]$ be the matrix formed by these eigenvectors. We shall diagonalize C by diagonalizing each of its four individual blocks with a similarity transformation using $\mathscr{V}$. We need a general solution for the discrete biharmonic equations on the subdomains, which is obtained by using the method of separation of variables.

Substituting the expression $V(ih, kh) = d_k\sqrt{4h}\sin(2ij\pi h)$ into the discrete model

$$\Delta_h^2 V + \lambda h^2 DV = 0, \tag{2.11}$$

we get the following fourth-order difference equation for d_k:

$$b_0 d_{k+2} + b_1 d_{k+1} + b_2 d_k + b_1 d_{k-1} + b_0 d_{k-2} = 0, \tag{2.12}$$

where

$$\begin{aligned} b_0(j) &= 1, \\ b_1(j) &= -8 + 4\cos(2j\pi h), \\ b_2(j) &= 20 - 2\lambda h^2 + (2\lambda h^2 - 16)\cos(2j\pi h) + 2\cos(4j\pi h). \end{aligned}$$

We denote by d_k the solution of (2.12) with boundary conditions

$$d_0 = 0, \quad d_1 = 1, \quad d_{m_1} = 0, \quad d_{m_1+1} = -d_{m_1-1}. \tag{2.13}$$

The boundary conditions (2.13) are needed for the computation of terms such as $A_{31}A_{11}^{-1}A_{13}v_j$ and $A_{14}^{\mathrm{T}}A_{11}^{-1}A_{14}v_j$. Similarly we denote by $\tilde{d}_k$ the solution of (2.12) with boundary conditions

$$\tilde{d}_0 = 0, \quad \tilde{d}_1 = 1, \quad \tilde{d}_{m_2-1} = \tilde{d}_{m_2+1}, \quad \tilde{d}_{m_2-2} = \tilde{d}_{m_2+2}. \tag{2.14}$$

The boundary conditions (2.14) are needed for the computation of terms such as $A_{32}A_{22}^{-1}A_{23}v_j$ and $A_{42}A_{22}^{-1}A_{24}v_j$. Explicit expressions for d_k and $\tilde{d}_k$ will be given later. By routine computation one may readily verify that the following relations hold:

$$A_{33}v_j = b_2 v_j,$$

$$A_{31}A_{11}^{-1}A_{13}v_j = \frac{1}{\alpha_j}(b_1^2 d_1 + 2b_1 d_2 + d_1)v_j,$$

$$A_{32}A_{22}^{-1}A_{23}v_j = \frac{1}{\beta_j}\tilde{d}_1 v_j,$$

$$A_{34}v_j = b_1 v_j,$$

$$A_{31}A_{11}^{-1}A_{14}v_j = \frac{1}{\alpha_j}(b_1 d_1 + d_2)v_j,$$

$$A_{23}^{\mathrm{T}}A_{22}^{-1}A_{13}v_j = \frac{1}{\beta_j}(b_1 \tilde{d}_1 + \tilde{d}_2)v_j,$$

$$A_{14}^{\mathrm{T}}A_{11}^{-1}A_{13}v_j = \frac{1}{\alpha_j}(b_1 d_1 + b_2)v_j,$$

$$A_{42}A_{22}^{-1}A_{23}v_j = \frac{1}{\beta_j}(b_1 \tilde{d}_1 + \tilde{d}_2)v_j,$$

$$A_{44}v_j = b_2 v_j,$$

$$A_{14}^{\mathrm{T}}A_{11}^{-1}A_{14}v_j = \frac{1}{\alpha_j}d_1 v_j,$$

$$A_{42}A_{22}^{-1}A_{24}v_j = \frac{1}{\beta_j}(b_1^2 \tilde{d}_1 + 2b_1\tilde{d}_2 + \tilde{d}_1)v_j,$$

where

$$\alpha_j = b_2(j) + b_1(j)d_2(m_1) + d_3(m_1),$$

$$\beta_j = b_2(j) + b_1(j)\tilde{d}_2(m_2) + \tilde{d}_3(m_2).$$

We note that in the computation of terms such as $A_{31}A_{11}^{-1}A_{13}v_j$ and $A_{42}A_{22}^{-1}A_{24}v_j$ and so on, only the first two components of $A_{31}v_j$ and $A_{13}v_j$ are nonzero. Thus, we only need to compute the first leading 2×2 block principal submatrices of A_{11}^{-1} and A_{22}^{-1}. In order to keep the middle term of the right-hand side of (2.15) symmetric, we make some simplification so that the leading 2×2 block principal submatrices of A_{11}^{-1} and A_{22}^{-1} are symmetric. We summarize these results by stating the following:

Theorem 2.1. *The interface Schur complement C has the following diagonalized form:*

$$C = \begin{bmatrix} \mathscr{V} & 0 \\ 0 & \mathscr{V} \end{bmatrix} \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix} \begin{bmatrix} \mathscr{V} & 0 \\ 0 & \mathscr{V} \end{bmatrix}, \tag{2.15}$$

where

$$\Lambda_{11} = \mathrm{diag}(\lambda_{11,j}), \quad \Lambda_{22} = \mathrm{diag}(\lambda_{22,j}), \quad \Lambda_{12} = \mathrm{diag}(\lambda_{12,j}), \quad \Lambda_{21} = \Lambda_{12},$$

and

$$\lambda_{11,j} = b_2 - \frac{1}{\alpha_j}(b_1^2 d_1(m_1) + 2b_1 d_2(m_1) + d_1(m_1)) - \frac{1}{\beta_j}\tilde{d}_1(m_2),$$

$$\lambda_{12,j} = b_1 - \frac{1}{\alpha_j}(b_1 d_1(m_1) + d_2(m_2)) - \frac{1}{\beta_j}(b_1\tilde{d}_1(m_2) + \tilde{d}_2(m_2)),$$

$$\lambda_{22,j} = b_2 - \frac{1}{\alpha_j} d_1(m_1) - \frac{1}{\beta_j}(b_1^2 \tilde{d}_1(m_1) + 2b_1\tilde{d}_2(m_2) + \tilde{d}_1(m_2)).$$

We now proceed to derive explicit expressions for d_k and $\tilde{d}_k$. The characteristic roots η for the difference scheme (2.12) satisfy the following relation:

$$b_0(\eta + \eta^{-1})^2 + b_1(\eta + \eta^{-1}) + b_2 - 2b_0 = 0,$$

or

$$\eta + \eta^{-1} = \frac{1}{2b_0}\left(- b_1 \pm \sqrt{b_1^2 + 8b_0^2 - 4b_2 b_0}\right) \equiv \frac{-b_1}{2} \pm \frac{\delta(j)}{2} \tag{2.16}$$

with

$$\delta(j) = \sqrt{b_1^2 + 8b_0^2 - 4b_2 b_0} = 4\sqrt{\lambda} h \sin(j\pi h).$$

Eq. (2.16) has four roots r, r^{-1}, s and s^{-1}, with

$$r = \frac{1}{4}\left(- b_1 + \delta - \sqrt{(b_1 - \delta)^2 - 16}\right)$$

and

$$s = \frac{1}{4}\left(- b_1 - \delta - \sqrt{(b_1 + \delta)^2 - 16}\right).$$

A straightforward computation yields the following solutions

$$\begin{aligned} d_k(m_1) = {} & \{k(r^{m_1} + r^{-m_1})(r^{k-m_1} - r^{m_1-k}) + 2(m_1 - k)(r^k - r^{-k})\} \\ & \times \{(r^{m_1} + r^{-m_1})(r^{1-m_1} - r^{m_1-1}) + 2(m_1 - 1)(r - r^{-1})\}^{-1}, \end{aligned}$$

and

$$\begin{aligned} \tilde{d}_k(m_2) = {} & \{k(r^{m_2} + r^{-m_2})(r^{k-m_2} - r^{m_2-k}) - 2m_2(r^k - r^{-k})\} \\ & \times \{(r^{m_2} + r^{-m_2})(r^{1-m_2} - r^{m_2-1}) - 2m_2(r - r^{-1})\}^{-1}. \end{aligned}$$

If we consider $0 < r, s < 1$, and let $m_1, m_2 \to \infty$ (i.e., in the case of an infinite strip), then

$$d_k = kr^{k-1}, \quad \tilde{d}_k = kr^{k-1}.$$

On the other hand, if we consider $0 < \frac{1}{r}, \frac{1}{s} < 1$ and let $m_1, m_2 \to \infty$, then

$$d_k = kr^{1-k}, \quad \tilde{d}_k = kr^{1-k}.$$

The results in Theorem 2.1 now take the form of:

Theorem 2.2. *For the discretization scheme of* (2.3) *on an infinite strip* ($m_1, m_2 = \infty$), *when* $\langle \mathrm{i} \rangle$ $0 < r < 1$, we have

$$\alpha_j = \beta_j = b_2(j) + 2b_1(j)r + 3b_0(j)r^2,$$

$$\lambda_{11,j} = b_2 - \frac{1}{\alpha_j}(b_1^2 + 4b_1 r + 2),$$

$$\lambda_{12,j} = b_1 - \frac{2}{\alpha_j}(b_1 + 2r),$$

and

$$\lambda_{22,j} = \lambda_{11,j};$$

and when $\langle \text{ii} \rangle$ $0 < 1/r < 1$, we have

$$\alpha_j = \beta_j = b_2(j) + 2b_1(j)\frac{1}{r} + 3b_0(j)\frac{1}{r^2},$$

$$\lambda_{11,j} = b_2 - \frac{1}{\alpha_j}\left(b_1^2 + 4b_1\frac{1}{r} + 2\right),$$

$$\lambda_{12,j} = b_1 - \frac{2}{\alpha_j}\left(b_1 + 2\frac{1}{r}\right),$$

and

$$\lambda_{22,j} = \lambda_{11,j}.$$

Now we are ready to construct preconditioners for the interface Schur complements. Denote by M and M_∞ the matrices obtained in Theorems 2.1 and 2.2, respectively. It is clear from their construction that M and M_∞ are respective analogs of Chan's [1] and Golub/Mayer's [10] preconditioners for fourth-order plate problems. Both of the preconditioners M and M_∞ will be implemented to solve the linear system in Step 3 of Algorithm 2.2.

2.2. The interface probing preconditioner

We observe that the interface operator C has strong spatial local coupling and weak global coupling in each continuation step, i.e., the entries of C decay rapidly away from the main diagonal, see Fig. 2. Hence, we use the interface probing technique to construct efficient preconditioners for C. The probing technique was introduced by Chan and Resasco [4], and by Keyes and Gropp [13], as an algebraic technique for constructing sparse approximation to the interface operator C. The main idea is to approximate C by a matrix having a specified sparsity pattern using matrix–vector products of C with a few carefully chosen probe vectors. The sparsity pattern is chosen to capture the strongest coupling of C.

Let M_d be a banded approximation with upper and lower bandwidth d to the interface operator C as proposed in [3]. We introduce the notation

$$M_d = \text{PROBE}(C, d) \tag{2.17}$$

to denote that M_d is constructed from C using the PROBE procedure, and let

$$M_{k,l} = \begin{bmatrix} \text{PROBE}(C_{11}, k) & \text{PROBE}(C_{12}, l) \\ \text{PROBE}(C_{21}, l) & \text{PROBE}(C_{22}, k) \end{bmatrix} \tag{2.18}$$

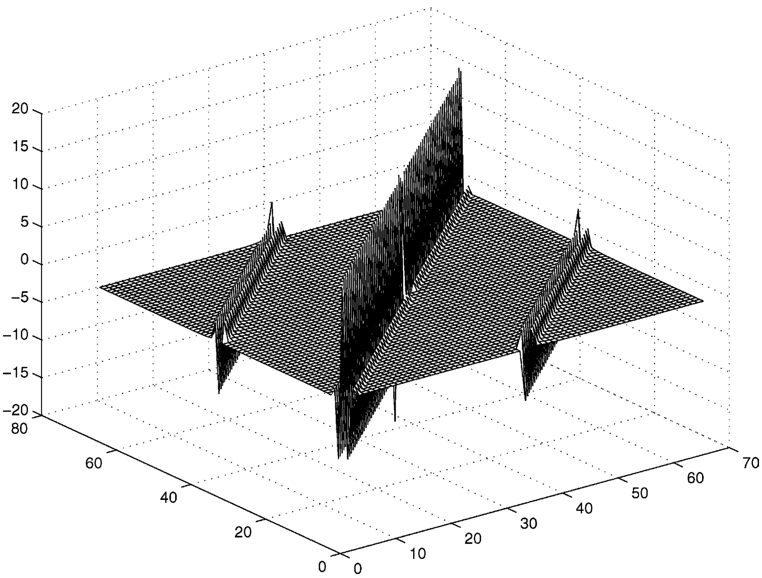

Fig. 2. Plot of elements of C.

be a preconditioner for C consisting of k-diagonal approximations for the diagonal blocks C_{11} and C_{22}, and l-diagonal approximations for the off-diagonal blocks C_{12} and C_{21}. We illustrate the PROBE procedure for the case $d = 1$, in which case M_1 is a tridiagonal matrix, and the following three probe vectors are given by: $v_1 = (1, 0, 0, 1, 0, 0, \ldots)^{\mathrm{T}}$, $v_2 = (0, 1, 0, 0, 1, 0, \ldots)^{\mathrm{T}}$ and $v_3 = (0, 0, 1, 0, 0, 1, \ldots)^{\mathrm{T}}$. Since M_1 is tridiagonal, it can be easily checked that all its nonzero entries appear in the vectors $M_1 v_i,\ i = 1, 2, 3$, as illustrated below:

$$\begin{bmatrix} m_{11} & m_{12} & & & & \\ m_{21} & m_{22} & m_{23} & & & \\ & m_{32} & m_{33} & m_{34} & & \\ & & m_{43} & m_{44} & m_{45} & \\ & & & m_{54} & m_{55} & \ddots \\ & & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & m_{23} \\ m_{34} & m_{32} & m_{33} \\ m_{44} & m_{45} & m_{43} \\ m_{54} & m_{55} & m_{56} \\ \vdots & \vdots & \vdots \end{bmatrix}. \tag{2.19}$$

The probe algorithm reconstructs the nonzero entries of M_1 by equating the right-hand side of (2.19) to the corresponding entries in the vectors $[Cv_1, Cv_2, Cv_3]$.

3. Domain decomposition for nonlinear problems

In this section, we discuss how to use both nonoverlapping and overlapping domain decomposition to solve fourth-order nonlinear plate problems.

3.1. Nonoverlapping domain decomposition

Theoretical study of domain decomposition method for nonlinear problems was given in [9]. We consider the reduced problem for the nonlinear von Kármán equations [8] as follows:

$$
\begin{aligned}
&\Delta^2 w+\lambda\frac{\partial^2 w}{\partial x^2}-[f,w]=0,\\
&\Delta^2 f+\tfrac{1}{2}[w,w]=0 \quad \text{in } \Omega=[0,\tfrac{1}{2}]\times[0,\tfrac{1}{2}],\\
&f=\Delta f=0,\quad w=\Delta w=0 \quad \text{on } x=0 \text{ and } y=0,\\
&f_n=f_{nnn}=0,\quad w_n=w_{nnn}=0 \quad \text{on } x=\tfrac{1}{2} \text{ and } y=\tfrac{1}{2}.
\end{aligned}
\tag{3.1}
$$

Let Ω be decomposed as in Section 2. Then the matrices A and D can be expressed as in Eqs. (2.5) and (2.6). Let E and $V\in\mathbb{R}^{L^2\times L^2}$ be the discretization matrices corresponding to the differential operators $\partial^2/\partial y^2$ and $\partial^2/\partial y\partial x$, respectively. We have

$$
E=\frac{1}{h^2}\left[\begin{array}{ccccc|ccccc|c|c}
-2I & I & & & & & & & & & I & \\
I & -2I & I & & & & & & & & & \\
 & \ddots & \ddots & \ddots & & & & & & & & \\
 & & \ddots & \ddots & I & & & & & & & \\
 & & & I & -2I & & & & & & & \\
\hline
 & & & & & -2I & I & & & & & I \\
 & & & & & I & -2I & I & & & & \\
 & & & & & & \ddots & \ddots & \ddots & & & \\
 & & & & & & & I & -2I & I & & \\
 & & & & & & & & 2I & -2I & & \\
\hline
I & & & & & & & & & & -2I & I \\
\hline
 & & & & & I & & & & & I & -2I
\end{array}\right]\in\mathbb{R}^{L^2\times L^2}
$$

and

$$
V=\frac{1}{4h^2}\left[\begin{array}{ccccc|ccccc|c|c}
0 & -V_L & & & & & & & & & V_L & \\
V_L & 0 & -V_L & & & & & & & & & \\
 & \ddots & \ddots & \ddots & & & & & & & & \\
 & & V_L & 0 & -V_L & & & & & & & \\
 & & & V_L & 0 & & & & & & & \\
\hline
 & & & & & 0 & V_L & & & & & -V_L \\
 & & & & & -V_L & 0 & V_L & & & & \\
 & & & & & & \ddots & \ddots & \ddots & & & \\
 & & & & & & & -V_L & 0 & V_L & & \\
 & & & & & & & & 0 & 0 & & \\
\hline
-V_L & & & & & & & & & & 0 & V_L \\
\hline
 & & & & & V_L & & & & & -V_L & 0
\end{array}\right]\in\mathbb{R}^{L^2\times L^2}
$$

where I is the $L\times L$ identity matrix, and

$$
V_L=\frac{1}{h^2}\begin{bmatrix}
0 & 1 & & & \mathbf{0} \\
-1 & 0 & 1 & & \\
 & \ddots & \ddots & \ddots & \\
 & & -1 & 0 & 1 \\
\mathbf{0} & & & 0 & 0
\end{bmatrix}\in\mathbb{R}^{L\times L}.
$$

Definition 3.1. For any $x = (x_1, \dots, x_N)^{\mathrm{T}}$, $y = (y_1, \dots, y_N)^{\mathrm{T}} \in \mathbb{R}^N$, we define $x * y \in \mathbb{R}^N$ by $x * y = (x_1 y_1, \dots, x_N y_N)^{\mathrm{T}}$. For any $A = (a_1, \dots, a_N)^{\mathrm{T}} \in \mathbb{R}^{N \times N}$ with a_i^{T} denotes the ith row of A, and $x = (x_1, \dots, x_N)^{\mathrm{T}} \in \mathbb{R}^N$, we define $A * x \in \mathbb{R}^{N \times N}$ by $A * x = (x_1 a_1, \dots, x_N a_N)^{\mathrm{T}}$.

Let $Z = [W, F]^{\mathrm{T}} \in \mathbb{R}^{2L^2}$, the discretization system corresponding to Eq. (3.1) is

$$H(Z, \lambda) = [H_1(Z, \lambda), H_2(Z, \lambda)]^{\mathrm{T}} = 0, \tag{3.2}$$

with

$$\begin{aligned} H_1(Z, \lambda) &= AW + \lambda DW - (DF) * (EW) + 2\left(\frac{1}{4} VF\right) * \left(\frac{1}{4} VW\right) - (EF) * (DW) \\ &= AW - (DW) * (EF) - (EW) * (DF) + \frac{1}{8}(VW) * (VF) + \lambda DW, \end{aligned}$$

$$H_2(Z, \lambda) = AF + (DW) * (EW) - \frac{1}{16}(VW) * (VW).$$

The Jacobian matrix corresponding to (3.2) is

$$DH(Z, \lambda) = [D_Z H(Z, \lambda), D_\lambda H(Z, \lambda)] = \begin{bmatrix} \tilde{A}(Z, \lambda) & -M(Z, \lambda) & DW \\ M(Z, \lambda) & A & 0 \end{bmatrix}, \tag{3.3}$$

where

$$M(Z, \lambda) = D * (EW) + E * (DW) - \frac{1}{8} V * (VW)$$

and

$$\tilde{A}(Z, \lambda) = A - D * (EF) - E * (DF) + \frac{1}{8} V * (VF) + \lambda D.$$

We refer to [6] for details.

3.2. The interface probing preconditioner for nonlinear systems

We rewrite the linear systems in Step 1 of Algorithm 2.1 as

$$\bar{A} z = b, \tag{3.4}$$

where $\bar{A} = D_Z H(y_i) = \begin{bmatrix} \tilde{A}(y_i) & -M(y_i) \\ M(y_i) & A \end{bmatrix}$. The structure of $\bar{A}$ is shown in Fig. 3. We can find a transformation matrix P such that Eq. (3.4) can be written as

$$\hat{A} u = p \tag{3.5}$$

with $\hat{A} = P^{\mathrm{T}} \bar{A} P$, $u = P^{\mathrm{T}} z$ and $p = P^{\mathrm{T}} b$. The structure of $\hat{A}$ is shown in Fig. 4. Eq. (3.5) can be expressed in block form as:

$$\begin{bmatrix} A_{11} & 0 & A_{13} & 0 & A_{15} \\ 0 & A_{22} & 0 & A_{24} & A_{25} \\ -A_{13} & 0 & A_{33} & 0 & A_{35} \\ 0 & -A_{24} & 0 & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}. \tag{3.6}$$

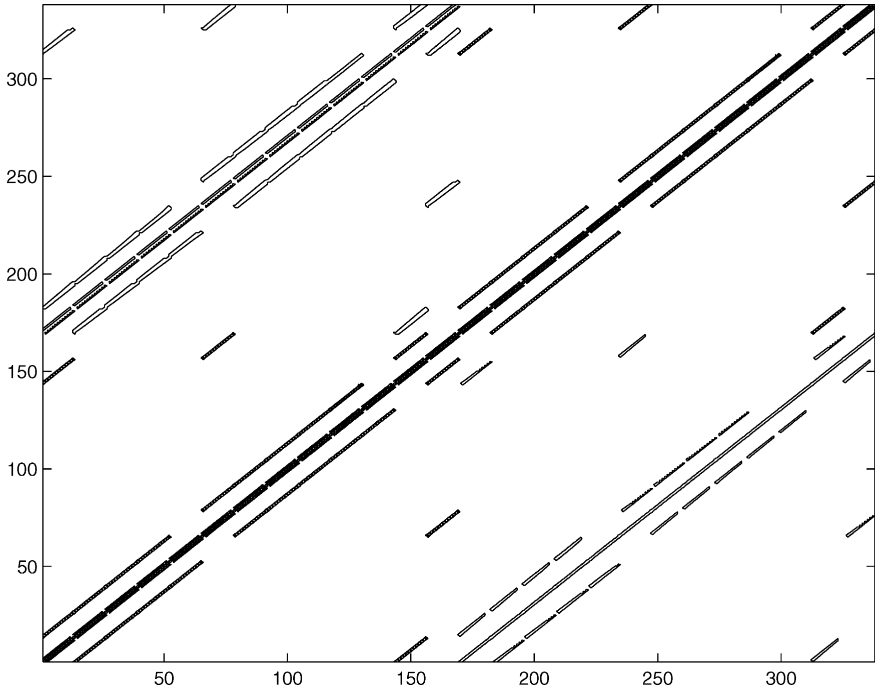

Fig. 3. The structure of $\bar{A}$.

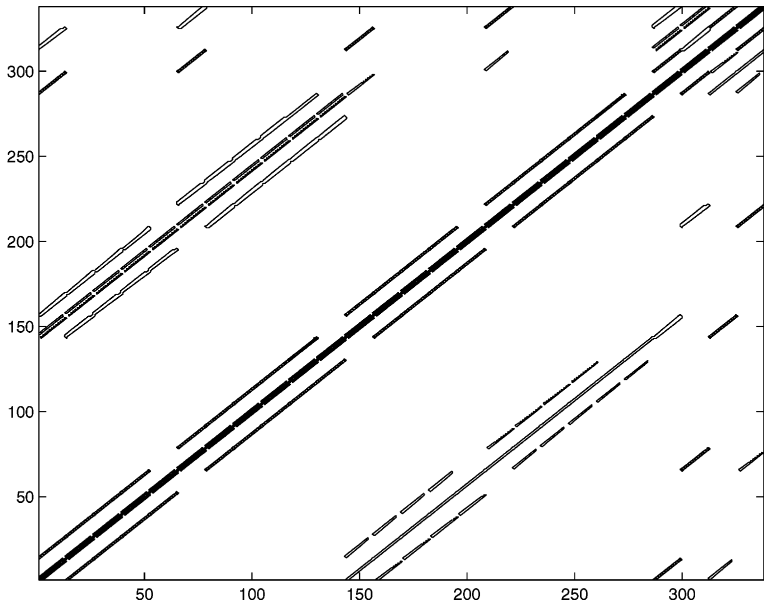

Fig. 4. The structure of $\hat{A}$.

From the linear system (3.6) we obtain $u_1 = A_{11}^{-1}(p_1 - A_{13}u_3 - A_{15}u_5)$, $u_2 = A_{22}^{-1}(p_2 - A_{24}u_4 - A_{25}u_5)$, $u_3 = A_{33}^{-1}(p_3 + A_{13}u_1 - A_{35}u_5)$, and $u_4 = A_{44}^{-1}(p_4 + A_{24}u_2 - A_{45}u_5)$. Substituting u_1, u_2, u_3, and u_4 into (3.6) and letting

$$E_1 = A_{11} + A_{13}A_{33}^{-1}A_{13},$$

$$E_2 = A_{22} + A_{24}A_{44}^{-1}A_{24},$$

$$E_3 = A_{33} + A_{13}A_{11}^{-1}A_{13},$$

$$E_4 = A_{44} + A_{24}A_{22}^{-1}A_{24},$$

$$F_1 = E_1^{-1}(-A_{15} + A_{13}A_{33}^{-1}A_{35}),$$

$$F_2 = E_2^{-1}(-A_{25} + A_{24}A_{44}^{-1}A_{45}),$$

$$F_3 = E_3^{-1}(A_{35} + A_{13}A_{11}^{-1}A_{15}),$$

$$F_4 = E_4^{-1}(A_{45} + A_{24}A_{22}^{-1}A_{25}),$$

$$f_1 = E_1^{-1}(p_1 - A_{13}A_{33}^{-1}p_3),$$

$$f_2 = E_2^{-1}(p_2 - A_{24}A_{44}^{-1}p_4),$$

$$f_3 = E_3^{-1}(p_3 + A_{13}A_{11}^{-1}p_1),$$

and

$$f_4 = E_4^{-1}(p_4 + A_{24}A_{22}^{-1}p_2)$$

we obtain the following reduced system

$$\bar{C}u_5 = \bar{p}, \tag{3.7}$$

where $\bar{C} = A_{55} + A_{51}F_1 + A_{52}F_2 - A_{53}F_3 - A_{54}F_4$ is the Schur complement matrix corresponding to the reduced interface operator, and $\bar{p} = p_5 - A_{51}f_1 - A_{52}f_2 - A_{53}f_3 - A_{54}f_4$. Hence, the solution to the linear system (3.6) is obtained by applying the block elimination again.

Algorithm 3.1 (*Block elimination algorithm for solving* (3.6)).

Step 1.	Solve	(1) $A_{33}X_1 = A_{13}, A_{33}Y_1 = A_{35}, A_{33}s_1 = p_3$,
		(2) $A_{44}X_2 = A_{24}, A_{44}Y_2 = A_{45}, A_{44}s_2 = p_4$,
		(3) $A_{11}X_3 = A_{13}, A_{11}Y_3 = A_{15}, A_{11}s_3 = p_1$,
	and	(4) $A_{22}X_4 = A_{24}, A_{22}Y_4 = A_{25}, A_{22}s_4 = p_2$.
Step 2.	Compute	(1) $E_1 = A_{11} + A_{13}X_1, Z_1 = -A_{15} + A_{13}Y_1, t_1 = p_1 - A_{13}s_1$,
		(2) $E_2 = A_{22} + A_{24}X_2, Z_2 = -A_{25} + A_{24}Y_2, t_2 = p_2 - A_{24}s_2$,
		(3) $E_3 = A_{33} + A_{13}X_3, Z_3 = A_{35} + A_{13}Y_3, t_3 = p_3 + A_{13}s_3$,
	and	(4) $E_4 = A_{44} + A_{24}X_4, Z_4 = A_{45} + A_{24}Y_4, t_4 = p_4 + A_{24}s_4$.
Step 3.	Solve	(1) $E_1F_1 = Z_1, E_1f_1 = t_1$,
		(2) $E_2F_2 = Z_2, E_2f_2 = t_2$,

		(3) $E_3F_3 = Z_3, E_3f_3 = t_3$,
	and	(4) $E_4F_4 = Z_4, E_4f_4 = t_4$.
Step 4.	Compute	$\bar{C} = A_{55} + A_{51}F_1 + A_{52}F_2 - A_{53}F_3 - A_{54}F_4$,
	and	$\bar{p} = p_5 - A_{51}f_1 - A_{52}f_2 - A_{53}f_3 - A_{54}f_4$.
Step 5.	Solve	$\bar{C}u_5 = \bar{p}$.
Step 6.	Compute	(1) $u_1 = f_1 + F_1u_5$, (2) $u_2 = f_2 + F_2u_5$,
		(3) $u_3 = f_3 - F_3u_5$, and (4) $u_4 = f_4 - F_4u_5$.

In Steps 1, 2, 3 and 6 of Algorithm 3.1, we can solve (1), (2), (3) and (4) on a parallel computer.

The Schur complement matrix $\bar{C}$ is dense, unsymmetric and expensive to form explicitly. However, it has strong spatial local coupling and weak global coupling in each continuation step. See Fig. 5. Hence, the interface probing preconditioners M_1, M_2, $M_{3,1}$, and $M_{5,3}$ defined in Section 2.2 will be used to solve the linear system in Step 5 of Algorithm 3.1.

3.3. Overlapping domain decomposition

In this section, we describe the overlapping domain decomposition for solving fourth-order plate problems.

3.3.1. Discretization of the von Kármán equation on the overlapping domain

We suppose that Ω is decomposed into s-fold overlapping subdomains $\Omega_1, \Omega_2, \dots, \Omega_s$ with $\Gamma_{i,i+1}$ as the common border of Ω_i and Ω_{i+1} as shown in Fig. 6.

We suppose that a uniform mesh with size h is used on Ω with L grid points on the y-axis. We also suppose that $\Omega^h_{i,i+1}$ and $\Omega^h_{i+1,i}$ are the mesh lines in Ω_i and Ω_{i+1}, respectively, which are nearest the mesh line $\Gamma^h_{i,i+1}$ of $\Gamma_{i,i+1}$. Let Ω^h_i be the remaining meshes of Ω_i . Assume that

$$l_1 = (m_1 + 2)h, l_2 = (m_2 + 3)h, \dots, l_{s-1} = (m_{s-1} + 3)h, l_s = (m_s + 1)h,$$

where m_i denotes the number of grid points along the y-axis of the subdomain Ω^h_i. Note that $L = m_1 + m_2 + \cdots + m_s + 3(s - 1)$ and $l_1 + l_2 + \cdots + l_s = \frac{1}{2}$. Suppose that the unknowns are ordered in such

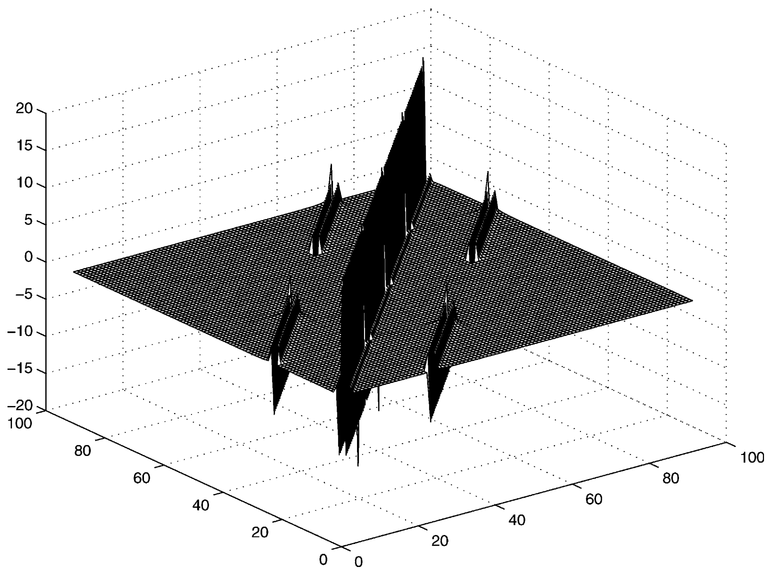

Fig. 5. Plot of elements of $\bar{C}$.

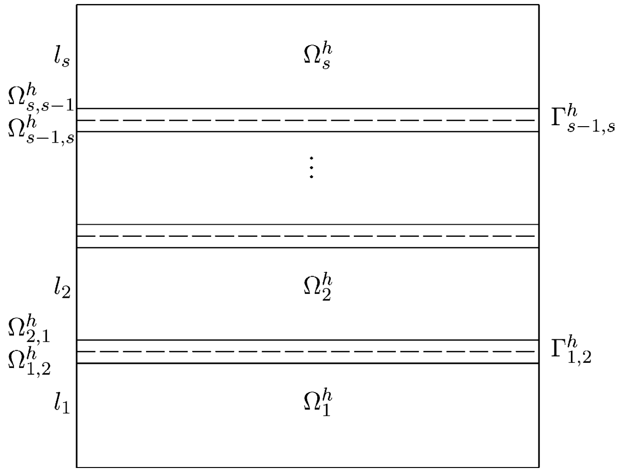

Fig. 6. The domain Ω and its partition for overlapping case.

a way that the interior points on $\Omega_1^h, \ldots, \Omega_s^h$ successively appear first, and those on the middle three grid lines that are enclosed by Ω_i^h and Ω_{i+1}^h appear last. Then the discrete solution vector w can be expressed as

$$w = \left[w_1, \ldots, w_s, w_{12}, w'_{12}, w_{21}, \ldots, w_{s-1,s}, w'_{s-1,s}, w_{s,s-1} \right],$$

where $w_i, w_{i,i+1}, w'_{i,i+1}$ and $w_{i+1,i}$s denote the unknowns on Ω_i^h, $\Omega_{i,i+1}^h$, $\Gamma_{i,i+1}^h$ and $\Omega_{i+1,i}^h$, respectively.

Let A, E, V and $D \in \mathbb{R}^{L^2 \times L^2}$ be the discretization matrices corresponding to the differential operators Δ^2, $\partial^2/\partial y^2$, $\partial^2/\partial y \partial x$ and $\partial^2/\partial x^2$, respectively. We have

$$A = \frac{1}{h^4} \left[\begin{array}{ccccc|ccccc|ccccc|ccc|c|ccc}
A_2 & B & I \\
B & A_1 & \ddots & \ddots \\
I & \ddots & \ddots & \ddots & I \\
 & \ddots & \ddots & A_1 & B & & & & & & & & & & & I \\
 & & I & B & A_1 & & & & & & & & & & & B & I \\
\hline
 & & & & & A_1 & B & I & & & & & & & & I & B \\
 & & & & & B & A_1 & B & I & & & & & & & & I \\
 & & & & & & & \ddots & & & & & & & & & & & \vdots \\
 & & & & & & I & B & A_1 & B & & & & & & & & & & I \\
 & & & & & & & I & B & A_1 & & & & & & & & & & B & I \\
\hline
 & & & & & & & & & & A_1 & B & I & & & & & & & & I & B \\
 & & & & & & & & & & B & A_1 & \ddots & \ddots & & & & & & & & I \\
 & & & & & & & & & & I & \ddots & \ddots & \ddots & I \\
 & & & & & & & & & & & \ddots & \ddots & A_2 & B \\
 & & & & & & & & & & & & 2I & 2B & A_1 \\
\hline
 & & & I & B & & & & & & & & & & & A_1 & B & I \\
 & & & & I & I & & & & & & & & & & B & A_1 & B \\
 & & & & & B & I & & & & & & & & & I & B & A_1 \\
\hline
 & & & & & & & \cdots & & & & & & & & & & & \ddots \\
\hline
 & & & & & & & & I & B & & & & & & & & & & A_1 & B & I \\
 & & & & & & & & & I & I & & & & & & & & & B & A_1 & B \\
 & & & & & & & & & & B & I & & & & & & & & I & B & A_1
\end{array} \right].$$

$$E = \frac{1}{h^2}\left[\begin{array}{c|c|c|c|c|c}
\begin{matrix} \bar{I} & I & & \\ I & \ddots & \ddots & \\ & \ddots & \ddots & I \\ & & I & \bar{I} \end{matrix} & & & \begin{matrix} & \\ & \\ & \\ I & \end{matrix} & & \\
\hline
& \begin{matrix} \bar{I} & I & & & \\ I & \bar{I} & I & & \\ & & \ddots & & \\ & I & \bar{I} & I & \\ & & I & \bar{I} & \end{matrix} & & \begin{matrix} & I \\ & \\ & \\ & \\ & \end{matrix} & \vdots & \begin{matrix} & \\ & \\ & \\ & \\ I & \end{matrix} \\
\hline
& & \begin{matrix} \bar{I} & I & & \\ I & \ddots & \ddots & \\ & \ddots & \ddots & I \\ & & -\bar{I} & \bar{I} \end{matrix} & & & \begin{matrix} & I \\ & \\ & \\ & \end{matrix} \\
\hline
\begin{matrix} & I \\ & \\ & \end{matrix} & \begin{matrix} & \\ & \\ I & \end{matrix} & & \begin{matrix} \bar{I} & I & \\ I & \bar{I} & I \\ & I & \bar{I} \end{matrix} & & \\
\hline
& \cdots & & & \ddots & \\
\hline
& \begin{matrix} & I \\ & \\ & \end{matrix} & \begin{matrix} & \\ & \\ I & \end{matrix} & & & \begin{matrix} \bar{I} & I & \\ I & \bar{I} & I \\ & I & \bar{I} \end{matrix}
\end{array}\right],$$

$$V = \frac{1}{4h^2}\left[\begin{array}{c|c|c|c|c|c}
\begin{matrix} 0 & V_L & & \\ \bar{V}_L & \ddots & \ddots & \\ & \ddots & \ddots & V_L \\ & & \bar{V}_L & 0 \end{matrix} & & & \begin{matrix} & \\ & \\ & \\ V_L & \end{matrix} & & \\
\hline
& \begin{matrix} 0 & V_L & & \\ \bar{V}_L & 0 & V_L & \\ & & \ddots & \\ & \bar{V}_L & 0 & V_L \\ & & \bar{V}_L & 0 \end{matrix} & & \begin{matrix} & \bar{V}_L \\ & \\ & \\ & \\ & \end{matrix} & \vdots & \begin{matrix} & \\ & \\ & \\ & \\ V_L & \end{matrix} \\
\hline
& & \begin{matrix} 0 & V_L & & \\ \bar{V}_L & \ddots & \ddots & \\ & \ddots & \ddots & V_L \\ & & 0 & 0 \end{matrix} & & & \begin{matrix} & \bar{V}_L \\ & \\ & \\ & \end{matrix} \\
\hline
\begin{matrix} & \bar{V}_L \\ & \\ & \end{matrix} & \begin{matrix} & \\ & \\ V_L & \end{matrix} & & \begin{matrix} 0 & V_L & \\ \bar{V}_L & 0 & V_L \\ & \bar{V}_L & 0 \end{matrix} & & \\
\hline
& \cdots & & & \ddots & \\
\hline
& \begin{matrix} & \bar{V}_L \\ & \\ & \end{matrix} & \begin{matrix} & \\ & \\ V_L & \end{matrix} & & & \begin{matrix} 0 & V_L & \\ \bar{V}_L & 0 & V_L \\ & \bar{V}_L & 0 \end{matrix}
\end{array}\right],$$

and $D = \mathrm{diag}(D_L, \ldots, D_L)$, where $A_i, B, I, V_L, D_L \in \mathbb{R}^{L\times L}$, for $i = 1, \ldots, 3$, and $\bar{I} = -2I$, $\bar{V}_L = -V_L$.

If we set $Z = [W, F]^{\mathrm{T}} \in \mathbb{R}^{2L^2}$, then the discretization system corresponding to Eq. (3.1) is

$$H(Z, \lambda) = [H_1(Z, \lambda), H_2(Z, \lambda)]^{\mathrm{T}} = 0, \tag{3.8}$$

where the formulae for $H_1(Z, \lambda)$, $H_2(Z, \lambda)$ and the Jacobian matrix are exactly the same as those given in Section 3.1.

3.3.2. The Schwarz alternating procedures

Now we discuss how to use the Schwarz alternating procedures to solve fourth-order plate problems such as the von Kármán equations. For simplicity, we rewrite the linear systems in Step 1 of Algorithm 2.1 as

$$\bar{A}u = b, \tag{3.9}$$

where $\bar{A} = D_Z H(y_i) = \begin{bmatrix} \tilde{A}(y_i) & -M(y_i) \\ M(y_i) & A \end{bmatrix}$. The structure of $\bar{A}$ is shown in Fig. 7. Evidently, the structures of $\tilde{A}(y_i)$ and $M(y_i)$ are the same as that of A. Since the von Kármán equations have two variables, each restriction operator R_i from Ω to Ω_i can be represented as

$$R_i = \left[\begin{array}{c|c} R_{ii} & 0 \\ \hline 0 & R_{ii} \end{array}\right], \quad i = 1, \ldots, s,$$

where

$$R_{11} = \left[\begin{array}{c|c|c|c|c|c|c|c} I_{m_1 L} & & & & & & & \\ \hline & & & & I_L & & & \\ \hline & & & & I_L & & & \end{array}\right] \in \mathbb{R}^{(m_1+2)L\times L^2},$$

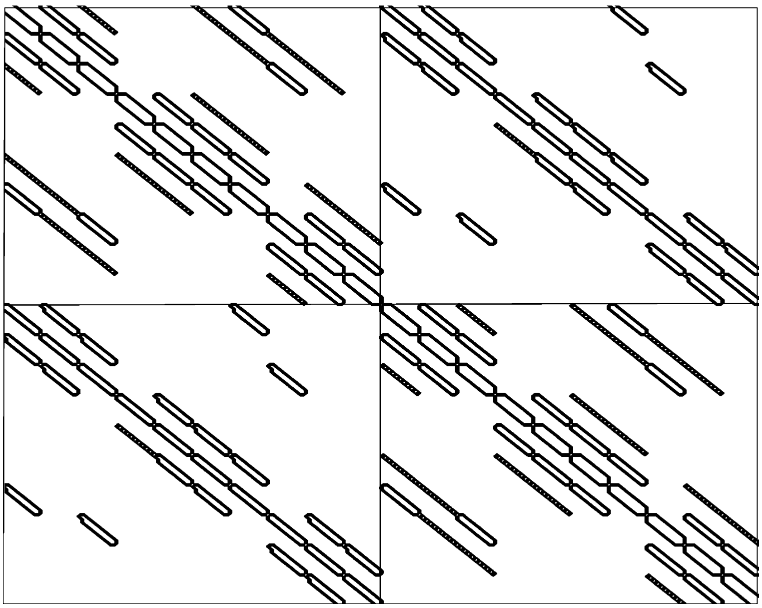

Fig. 7. The structure of $\bar{A}$ for Eq. (3.9).

$$R_{22} = \left[\begin{array}{c|c|c|c|c|c|c} & I_{m_2L} & & & & & \\ \hline & & & & I_L & & \\ \hline & & & & I_L & & \\ \hline & & & & & I_L & \\ \hline & & & & & I_L & \end{array}\right] \in \mathbb{R}^{(m_2+4)L\times L^2},$$

$\vdots$

and

$$R_{ss} = \left[\begin{array}{c|c|c|c|c|c|c} & & & I_{m_sL} & & & \\ \hline & & & & & & I_L \\ \hline & & & & & & I_L \end{array}\right] \in \mathbb{R}^{(m_s+2)L\times L^2}$$

with I_k denotes the $k \times k$ identity matrix. Furthermore, the upper left part of the matrix R_i corresponds to the function w, and the lower right part of R_i corresponds to the function f.

The transpose matrix R_i^{T} of R_i is a prolongation operator which takes a variable from Ω_i and extends it to a variable in Ω.

Define $\bar{A}_i = R_i\bar{A}R_i^{\mathrm{T}}$ of dimension $(2m_i+2)L \times (2m_i+2)L$ for $i=1$ or s and of dimension $(2m_i+4)L \times (2m_i+4)L$ for $i=2,\ldots,s-1$ as the restrictions of $\bar{A}$ to Ω_i . The multiplicative Schwarz method is described as follows:

Algorithm 3.2 (*Multiplicative Schwarz iteration*).
For $i=1,\ldots,s$ Do
$u = u + R_i^{\mathrm{T}}\bar{A}_i^{-1}R_i(b-\bar{A}u)$
EndDo

This method converges very slowly in practical numerical computation. Therefore we consider its preconditioned form. As described in [14], we define the operators $P_i = R_i^{\mathrm{T}}\bar{A}_i^{-1}R_i\bar{A}$, for $i=1,\ldots,s$ and $Q_s = (I-P_s)(I-P_{s-1})\cdots(I-P_1)$. We note that the multiplicative Schwarz method can be regarded as a fixed-point method for the following system

$$M_{\mathrm{ms}}^{-1}\bar{A}u = M_{\mathrm{ms}}^{-1}b, \tag{3.10}$$

in which $M_{\mathrm{ms}}^{-1}\bar{A} = I - Q_s, M_{\mathrm{ms}}^{-1}b = (I-Q_s)\bar{A}^{-1}b$. Clearly, M_{ms} plays the role of a preconditioner, and it is called the multiplicative Schwarz preconditioner.

Setting $T_i = P_i\bar{A}^{-1} = R_i^{\mathrm{T}}\bar{A}_i^{-1}R_i,\ i=1,\ldots,s$, the algorithm of the multiplicative Schwarz preconditioner becomes

Algorithm 3.3 (*Multiplicative Schwarz preconditioner*).
1. Input: b, output: $z = M_{\mathrm{ms}}^{-1}b$.
2. $\quad z = T_1 b$
3. For $i = 2 : s$ Do
4. $\quad z = z + T_i(b - \bar{A}z)$
5. EndDo

Algorithm 3.4 (*Multiplicative Schwarz preconditioned operator*).
1. Input: v, output: $z = M_{\mathrm{ms}}^{-1}\bar{A}v$.

2. $z = P_1 v$
3. For $i = 2, \ldots, s$ Do
4. $z = z + P_i(v - z)$
5. EndDo

Now we discuss how Algorithms 3.3 and 3.4 can be used to solve the preconditioned nonsymmetric linear system (3.10). First, the right-hand side of (3.10) can be computed by Algorithm 3.3. Then we can use some nonsymmetric linear solvers such as GMRES to solve (3.10), where the preconditioned operator $M_{\mathrm{ms}}^{-1}\bar{A}$ is computed by Algorithm 3.4.

The additive Schwarz procedure differs from the multiplicative one only on the components in which each subdomain is not updated until a whole cycle of updates through all subdomains are completed.

Algorithm 3.5 (*Additive Schwarz iteration*).
1. Compute $r_0 = b - \bar{A}u$
2. For $i = 1, \ldots, s$ Do
3. Compute $\delta_i = R_i^{\mathrm{T}}\bar{A}_i^{-1}R_i r_0$
4. EndDo
5. $u = u + \delta_1 + \delta_2 + \cdots + \delta_s$.

A similar equivalence relation can be stated between the additive Schwarz method and a generalized block-Jacobi iteration method, see [15]. Let M_{as} be the additive Schwarz preconditioner. The additive Schwarz procedure is described as follows:

Algorithm 3.6 (*Additive Schwarz preconditioner*).
1. Input: b, Output: $z = M_{\mathrm{as}}^{-1}b$.
2. For $i = 1, \ldots, s$ Do
3. Compute $z_i = T_i b$
4. EndDo
5. Compute $z = z_1 + z_2 + \cdots + z_s$.

Note that the do loop can be performed in parallel.

Algorithm 3.7 (*Additive Schwarz preconditioned operator*).
1. Input: v, Output: $z = M_{\mathrm{as}}^{-1}\bar{A}v$.
2. For $i = 1, \ldots, s$ Do
3. Compute $z_i = P_i v$
4. EndDo
5. Compute $z = z_1 + z_2 + \cdots + z_s$.

4. Numerical results

We used predictor–corrector continuation methods to trace the first solution branch of (3.1) bifurcating from the first bifurcation point $(u, \lambda_{1,1}) = (0, 4\pi^2) \approx (0, 39.478418)$. Both nonoverlapping and overlapping domain decomposition were considered. All of our computations were executed on an IBM Pentium 4 machine using MATLAB with double precision arithmetic.

The following notations are used in Tables 1–4.

NCS	ordering of the continuation steps.
ε	accuracy tolerance in Newton corrector.
κ_2	the two-norm condition number of $D_wH(y_i)$.
tol	stopping criterion for the GMRES method.
NCI	number of corrector iterations required at each continuation step.
MAXN	maximum norm of the approximating solution w.
itr	average number of iterations required by using the GMRES to solve linear systems in predictor (corrector) steps.
Time	the total execution time (in seconds) for performing 60 continuation steps.

Example 1 (*Nonoverlapping domain decomposition*). Eq. (3.1) was discretized by the centered difference approximations with uniform meshsize $h = 0.02174$ on the x- and y-axis, respectively. Let the two interface grid lines Ω_3^h and Ω_4^h be $y = 0.25 - (h/2)$ and $y = 0.25 + (h/2)$, respectively. We chose $m_1 = 10$, $m_2 = 11$, and obtained the linear system

$$\hat{A}u = p, \tag{4.1}$$

Table 1
Sample result for Example 1, $h = 0.02174$, $\varepsilon = 5.0 \times 10^{-4}$, tol $= 10^{-10}$, $\lambda^* = 39.3179$, overlapping subdomains

NCS	λ	MAXN	κ_2	Method	I	II	III	IV	V
				NCI	itr	itr	itr	itr	itr
6	38.4839	0.03706	3.0039E+07	2	92	63	82	61	77
11	39.1151	0.09865	7.4413E+07	2	92	64	83	61	79
16	39.3179	0.18009	9.9096E+07	2	92	64	83	61	80
21	39.4374	0.26264	8.6110E+07	2	92	64	83	61	80
26	39.5468	0.34512	6.4552E+07	2	92	64	82	61	80
31	39.6562	0.42701	5.0073E+07	2	92	64	82	61	80
41	47.1121	2.21021	1.7805E+07	2	92	63	80	60	79
60	127.9609	8.24570	4.6549E+05	2	92	60	78	59	78
				Time	1361	1350	1385	1346	1382

Table 2
Sample result for Example 1, $h = 0.01282$, $\varepsilon = 5.0 \times 10^{-4}$, tol $= 10^{-10}$, $\lambda^* = 39.3430$, overlapping subdomains

NCS	λ	MAXN	κ_2	Method	I	II	III	IV	V
				NCI	itr	itr	itr	itr	itr
8	38.8897	0.05102	4.0627E+08	3	156	95	44	91	41
18	39.3091	0.14811	9.0386E+08	2	156	97	45	94	43
20	39.3430	0.16802	9.2219E+08	2	156	97	45	94	43
22	39.3729	0.18793	9.1486E+08	2	156	97	45	94	43
31	39.4881	0.27752	7.1976E+08	2	156	97	45	94	43
41	39.8568	0.53486	2.7193E+08	2	156	95	44	94	43
51	40.4772	0.82702	1.2282E+08	2	156	94	44	93	42
60	41.2241	1.10838	7.4785E+07	2	156	92	44	92	42
				Time	37,701	33,498	30,594	31,113	30,358

Table 3
Sample result for Example 2, $h = 0.02174$, $\varepsilon = 5.0 \times 10^{-4}$, tol $= 10^{-10}$, $\lambda^* = 39.3179$, the domain Ω was decomposed into two and three overlapping subdomains

NCS	λ	MAXN	κ_2	Method	I	VI	VII
				NCI	itr	itr	itr
8	38.8164	0.05557	4.4593E+07	2	693, 693	21, 36	42, 69
14	39.2560	0.14720	9.4943E+07	2	782, 782	22, 37	43, 72
16	39.3179	0.18009	9.9096E+07	2	801, 801	23, 37	44, 72
18	39.3691	0.21309	9.6727E+07	2	806, 806	23, 37	44, 72
28	39.5921	0.37804	5.7012E+07	2	815, 815	23, 38	45, 73
38	39.8494	0.54172	3.1978E+07	2	820, 820	23, 38	45, 74
48	40.1693	0.70338	2.0044E+07	2	822, 822	23, 38	45, 74
60	56.9003	3.45530	1.4025E+06	2	856, 856	23, 38	46, 75
				Time	7220, 9406	2399, 2444	3580, 3594

Table 4
Sample result for Example 3, $h = 0.01724$, $\varepsilon = 5.0 \times 10^{-4}$, tol $= 10^{-10}$, $\lambda^* = 39.3919$, the domain Ω was decomposed into two, three, and four overlapping subdomains, respectively

NCS	λ	MAXN	κ_2	Method	I	VI	VII
				NCI	itr	itr	itr
8	39.0344	0.03377	1.6966E+08	3	1082, 1061, 1082	25, 44, 57	49, 86, 109
14	39.3558	0.10994	4.3828E+08	2	1190, 1115, 1190	27, 46, 62	52, 89, 117
16	39.3919	0.13648	4.5570E+08	2	1225, 1211, 1225	27, 46, 62	52, 90, 119
18	39.4217	0.16305	4.3912E+08	2	1247, 1230, 1247	27, 46, 63	53, 90, 122
28	39.6460	0.37512	1.6341E+08	2	1280, 1273, 1280	28, 47, 63	55, 92, 124
38	40.3410	0.76635	4.3862E+07	2	1288, 1281, 1288	28, 47, 64	55, 92, 124
48	41.4333	1.14396	2.0879E+07	2	1292, 1287, 1292	28, 47, 65	55, 92, 128
60	43.1923	1.57418	1.1886E+07	2	1309, 1293, 1309	28, 47, 65	55, 93, 128
				Time	25,035, 32,827, 33,498	11,088, 9252, 9244	16,066, 13,567, 12,748

where $\hat{A} \in \mathbb{R}^{1058\times1058}$. The Schur complement C is a matrix of order 92×92. We used the GMRES without preconditioner and the preconditioned GMRES method with preconditioners M_1, M_2, $M_{3,1}$ and $M_{5,3}$, which are denoted by the methods I, II, III, IV and V, respectively, to solve Step 5 of Algorithm 3.1. Table 1 shows our sample numerical results, where the first bifurcation point was detected at $(u^*, \lambda^*) = (0, 39.3179)$.

Next, we reduce the uniform meshsize to $h = 0.01282$ on the x- and y-axis, respectively, and implemented the same methods as above. We chose $m_1 = 18$, $m_2 = 19$, and obtained the linear system (4.1), where $\hat{A} \in \mathbb{R}^{3042\times3042}$. The Schur complement C is a matrix of order 156×156. Table 2 shows our sample numerical results, where the first bifurcation point was detected at $(u^*, \lambda^*) = (0, 39.3430)$. Compared to the average number of iterations required in each continuation step and the total execution time, the preconditioner $M_{5,3}$ is superior to the other preconditioners. Fig. 8 shows the convergence behavior of the methods I–V, where $h = 0.01282$.

Example 2 *(Overlapping domain decomposition)*. We exploited the numerical methods described in Section 3.3 to trace the first solution branch of the reduced problem (3.1). The domain Ω was decomposed into two overlapping subdomains Ω_1 and Ω_2. Eq. (3.1) was discretized by using the centered difference approximations with uniform meshsize $h = 0.02174$ on the x- and y-axis, respectively. Let the common border Γ be $y = 0.25$. We chose $m_1 = m_2 = 10$, and obtained the linear system (4.1), where $\hat{A} \in \mathbb{R}^{1058\times1058}$.

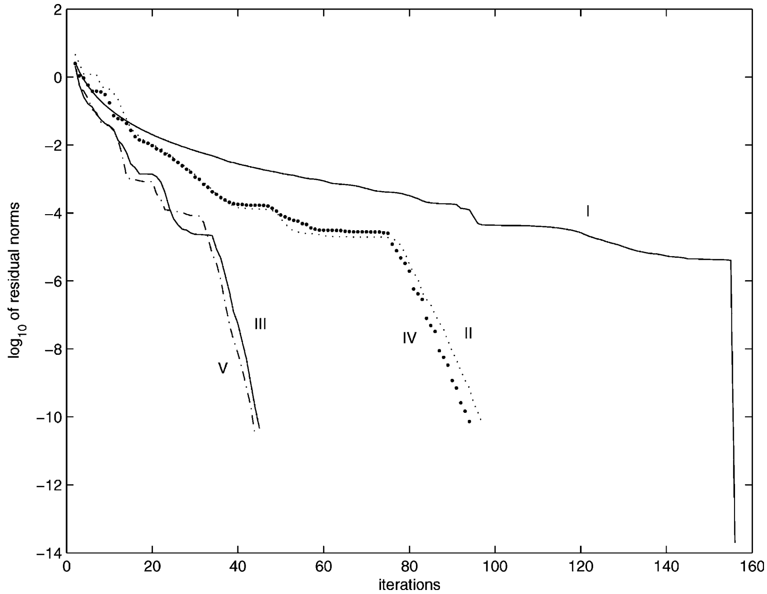

Fig. 8. Convergence behavior of the methods I–V at $\lambda = 39.3583$, where $h = 0.01282$.

The GMRES method, and the preconditioned GMRES method with preconditioners M_{ms} and M_{as}, which are denoted by the methods VI, VII, were implemented to trace the solution curve of Eq. (3.1). In total we executed 60 continuation steps, and 240 linear systems were solved. Next, the domain Ω was decomposed into three overlapping subdomains Ω_1, Ω_2 and Ω_3. Let $\Gamma_{1,2}$ and $\Gamma_{2,3}$ be $y = \frac{7}{46}$ and $y = \frac{8}{23}$, respectively. We chose $m_1 = 5$, $m_2 = m_3 = 6$, and obtained the linear system (4.1), where $\hat{A} \in \mathbb{R}^{1058 \times 1058}$. We

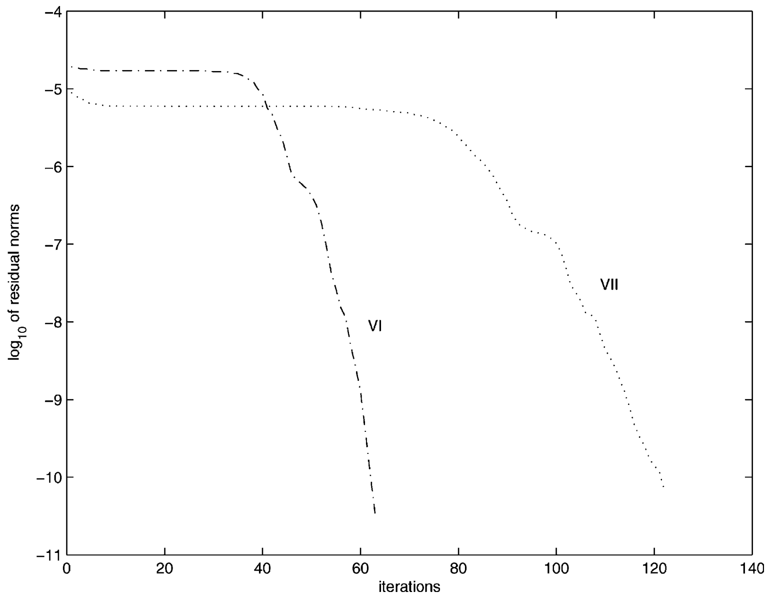

Fig. 9. Convergence behavior of the methods VI and VII on four subdomains at $\lambda = 39.4074$, where $h = 0.01724$.

used the same methods as above to trace the first solution curve of (3.1). In total we executed 60 continuation steps, and 240 linear systems were solved.

Table 3 shows that the bifurcation point was detected at $(u^*, \lambda^*) = (0, 39.3179)$.

Example 3 (*Overlapping domain decomposition*). We implemented the same methods as in Example 2 with uniform meshsize $h = 0.01724$ on the x- and y-axis, respectively. The domain Ω was decomposed into two, three, and four overlapping subdomains, respectively. In the case of two subdomains, let the interface Γ be $y = 0.25$ and we chose $m_1 = m_2 = 13$. In the case of three subdomains, let $\Gamma_{1,2}$ be $y = \frac{9}{58}$ and $\Gamma_{2,3}$ be $y = \frac{10}{29}$. We chose $m_1 = 7$, $m_2 = m_3 = 8$. In the case of four subdomains, let $\Gamma_{1,2}$, $\Gamma_{2,3}$ and $\Gamma_{3,4}$ be $y = \frac{7}{58}$, $y = \frac{15}{58}$ and $y = \frac{23}{58}$, respectively. We chose $m_1 = m_2 = m_3 = m_4 = 5$. In all cases, we obtained the linear systems (4.1), where $\hat{A} \in \mathbb{R}^{1682 \times 1682}$.

Tables 4 shows the numerical results, where the first bifurcation point was detected at $(u^*, \lambda^*) = (0, 39.3919)$. Fig. 9 shows the convergence behavior of the methods VI and VII on four subdomains, where $h = 0.01724$.

5. Conclusions

Based on the numerical results reported in Section 4, we wish to give some concluding remarks concerning the performance of the numerical algorithms we described in Sections 2 and 3.

1. For the nonoverlapping domain decomposition, the results in Tables 1 and 2 show that the methods II–V are effective only when the order of the coefficient matrix is large enough.
2. From Tables 1 and 3 we see that the nonoverlapping domain decomposition with various preconditioners is superior to the overlapping domain decomposition with multiplicative or additive preconditioner if the same size of linear systems are solved. For the overlapping domain decomposition, however, the preconditioned GMRES with multiplicative or additive preconditioner is superior to the GMRES.

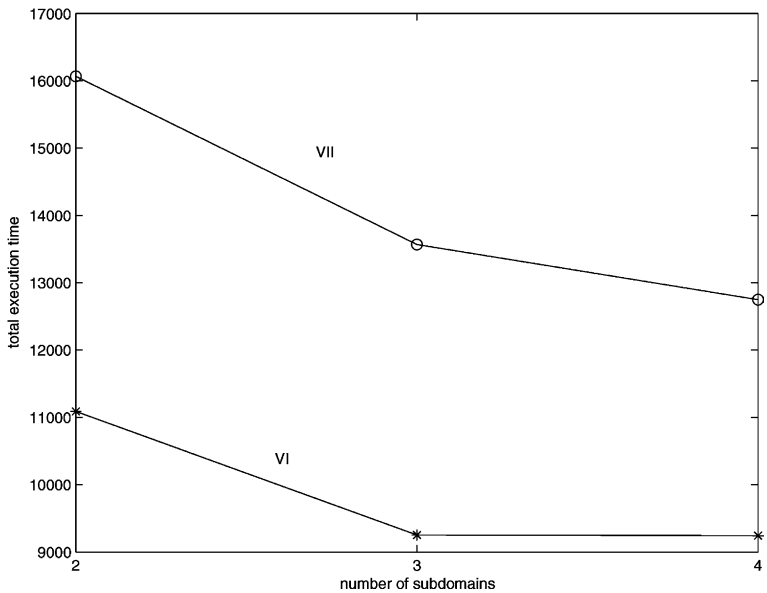

Fig. 10. Total execution time on two, three and four overlapping subdomains, where $h = 0.01724$.

3. For the overlapping domain decomposition, from the viewpoint of the average number of iterations required in each continuation step and the total execution time, the multiplicative Schwarz preconditioner is obviously superior to the additive Schwarz preconditioner, see Table 3. Moreover, for both cases, the total execution time decreases as the number of subdomains increases. See Fig. 10. It is also interesting to see that the average number of iterations required to solving linear systems increases as the number of subdomains decreases.

References

[1] T.F. Chan, Analysis of preconditioners for domain decomposition, SIAM J. Numer. Anal. 24 (1987) 382–390.
[2] T.F. Chan, T.P. Mathew, Domain decomposition algorithms, Acta Numer. (1994) 61–143.
[3] T.F. Chan, T.P. Mathew, The interface probing technique in domain decomposition, SIAM J. Matrix Anal. Appl. 13 (1992) 212–238.
[4] T.F. Chan, D.C. Resasco, A survey of preconditioners for domain decomposition, Technical Report/DCS/RR-414, Department of Computer Science, Yale University, New Haven, CT, 1985.
[5] T.F. Chan, E. Weinan, Jiachang Sun, Domain decomposition interface preconditioners for fourth-order elliptic problems, Appl. Numer. Math. 8 (1991) 317–331.
[6] C.-S. Chien, M.-S. Chen, Multiple bifurcations in the von Kármán equations, SIAM J. Sci. Comput. 18 (1997) 1737–1766.
[7] C.-S. Chien, S.-Y. Gong, Z. Mei, Mode jumping in the von Kármán equations, SIAM J. Sci. Comput. 22 (2000) 1354–1385.
[8] C.-S. Chien, Y.-J. Kuo, Z. Mei, Symmetry and scaling properties of the von Kármán equations, Z. Angew. Math. Phys. 49 (1998) 710–729.
[9] M. Dryja, W. Hackbusch, On the nonlinear domain decomposition method, BIT 37 (1997) 296–311.
[10] G.H. Golub, D. Mayers, The use of pre-conditioning over irregular regions, in: R. Glowinski, J.L. Lions (Eds.), Proceedings of the Sixth International Conference on Computing Methods in Applied Mechanics and Engineering, Versailles, France, 1984, pp. 3–14.
[11] M. Golubitsky, D.G. Schaeffer, Singularities and Groups in Bifurcation Theory, Vol. I, Springer, New York, 1984.
[12] H.B. Keller, Lectures on Numerical Methods in Bifurcation Problems, Springer, Berlin, 1987.
[13] D.E. Keyes, W.D. Gropp, A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation, SIAM J. Sci. Statist Comput. 8 (1987) 166–202.
[14] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Co., Boston, 1996.
[15] B.F. Smith, P.E. Bjorstad, W.D. Gropp, Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, Cambridge, UK, 1996.