FAST DIRECT SOLVERS FOR PIECEWISE HERMITE BICUBIC ORTHOGONAL SPLINE COLLOCATION EQUATIONS

BERNARD BIALECKI, GRAEME FAIRWEATHER AND KARIN R. BENNETT *

Abstract. Fast direct methods are proposed for the solution of linear systems arising when orthogonal spline collocation with piecewise Hermite bicubics is employed for the approximate solution of Poisson's equation in a rectangle. The new methods, which are matrix decomposition algorithms involving fast Fourier transforms, require $O(N^2 \log N)$ arithmetic operations on an $N \times N$ uniform partition and are highly parallel in nature. The treatment of more general elliptic problems is also discussed and the results of numerical experiments using parallel implementations of the new algorithms on several shared memory machines are presented.

Key Words. Orthogonal spline collocation, piecewise Hermite bicubics, generalized eigenvalue problem, eigenvalues and eigenfunctions, fast direct solvers, matrix decomposition algorithms, fast Fourier transforms, parallel implementations.

AMS(MOS) subject classification. 65F05, 65N30, 65N35, 65W05

1. Introduction. Over the past three decades, several special direct methods have been developed for the solution of elliptic partial difference equations on rectangular domains. The algorithms are usually discussed in terms of solving the linear systems associated with five-point finite difference operators although they are applicable to more general difference approximations such as the nine-point difference operator; see [18], for example. Developments in this area have been described by several authors, most notably Buzbee et al. [6], Dorr [10], and Swarztrauber [20]. More recently, Swarztrauber and Sweet [21] discussed current work in the context of parallel computing and provided a comprehensive bibliography. Software implementing fast direct methods is available in ELLPACK [18].

The primary purpose of this paper is to describe a new fast direct method for solving the linear systems arising when orthogonal spline collocation with piecewise Hermite bicubics defined on a uniform partition is applied to the homogeneous Dirichlet problem for Poisson's equation on a rectangle:

$$-\Delta u = f(x_1, x_2), \qquad (x_1, x_2) \in \Omega = (a_1, b_1) \times (a_2, b_2), \tag{1.1a}$$

* Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506, U.S.A.

(1.1b) $$u(x_1, x_2) = 0, \qquad (x_1, x_2) \in \partial\Omega.$$

(By a uniform partition of Ω, we mean a partition which is uniform in each of the coordinate directions but with possibly different stepsizes in these directions.) The new method has its roots in the work of Lynch et al. [15], where a procedure is given for solving the difference equations arising from the usual five-point stencil. In the case of an $N \times N$ uniform partition, these equations may be written in the form

(1.2) $$(T \otimes I + I \otimes T)\vec{u} = \vec{f},$$

where T is the tridiagonal matrix of order N associated with the standard second order difference operator, I is the $N \times N$ identity matrix, and $\otimes$ denotes the tensor product. Lynch et al. exhibit a diagonal matrix Λ and a symmetric matrix Z such that

(1.3) $$TZ = Z\Lambda, \qquad Z^2 = I.$$

Using (1.3), the system (1.2) may be rewritten in the form

(1.4) $$(\Lambda \otimes I + I \otimes \Lambda)\vec{v} = \vec{g},$$

where

(1.5) $$\vec{g} = (Z \otimes Z)\vec{f},$$

and

(1.6) $$\vec{u} = (Z \otimes Z)\vec{v}.$$

Since the coefficient matrix of the system (1.4) is diagonal, the vector $\vec{v}$ is easily determined. It is shown in [15] that the operation count for the procedure consisting of (1.5), (1.4), (1.6), which is commonly known as a matrix decomposition algorithm, is $O(N^3)$. However, if fast Fourier transforms (FFTs) are used to compute $\vec{g}$ and $\vec{u}$ in the algorithm of Lynch et al., then the operation count becomes $O(N^2 \log N)$.

When orthogonal spline collocation is applied to the boundary value problem (1.1) on an $N \times N$ uniform partition, the resulting system of equations is of the form

(1.7) $$(A \otimes B + B \otimes A)\vec{u} = \vec{f},$$

where the $2N \times 2N$ matrices A and B have a special structure called almost block diagonal, [5]. If a diagonal matrix Λ and matrix Z, each of order $2N$, can be determined so that

$$AZ = BZ\Lambda, \qquad Z^T B^T B Z = I, \tag{1.8}$$

then the system (1.7) can be written in a form similar to (1.4)–(1.6). Therefore, the problem is to determine Λ and Z, which is essentially equivalent to solving the generalized eigenvalue problem $A\vec{z} = \lambda B\vec{z}$. In this paper, we present matrices Λ and Z satisfying (1.8) for the case in which piecewise Hermite bicubics defined on a uniform partition are employed in the orthogonal spline collocation procedure.

A brief outline of this paper is as follows. In Section 2, we consider the piecewise Hermite cubic collocation approximation to the eigenvalue problem

$$-u''(x) = \lambda u(x), \quad x \in (a,b), \qquad u(a) = u(b) = 0, \tag{1.9}$$

from which we obtain the matrices Λ and Z. The matrix Λ was obtained in a slightly different fashion by Dyksen [12], who also gave expressions for the corresponding eigenfunctions in [11]. We give a systematic derivation which is easily generalized to other boundary conditions and to higher degree piecewise polynomial spaces. While our representation of the eigenvalues differs from that of Dyksen, these representations are, of course, equivalent. A key observation in our work is that the eigenfunctions can be "normalized" so that the matrix Z satisfies the second equality in (1.8). In Section 3, the new fast direct algorithm for solving the piecewise Hermite bicubic collocation equations is described along with its extension to problems more general than (1.1). The new algorithms may be viewed as orthogonal spline collocation analogues of the matrix decomposition method of Buzbee et al. [6]. In Section 4, we present a representative set of numerical examples solved using parallel implementations of the new algorithms on several shared memory machines. The numerical results not only demonstrate the efficacy of the parallelization but also exhibit superconvergence in the approximations to the first partial derivatives and the second order mixed partial derivative not previously reported in the literature. Further generalizations of the algorithms are discussed in Section 5.

It should be noted that Dyksen [12] used the $2N$ diagonal elements of the matrix Λ as iteration parameters in an alternating direction orthogonal spline collocation method for the solution of (1.1). The cost of Dyksen's algorithm regarded as a direct

method is, according to its author, $O(328N^3)$ operations. However he does show experimentally that convergence can be achieved with fewer than $2N$ iterations but no indication is given of how many one would perform to solve a given problem. This difficulty is overcome by Bialecki [3] who justifies the use of Jordan's acceleration parameters [22] in the alternating direction method and discusses ways of determining these. With this selection of parameters, which, it should be noted, requires only estimates of the smallest and largest eigenvalues, the number of iterations is proportional to $\log N$. Cooper and Prenter [7] examined the convergence of alternating direction collocation iteration applied to a large class of linear separable elliptic partial differential equations on a rectangular domain, but did not address the problem of determining optimal parameters.

The first algorithm described in this paper requires that the approximating space be the Hermite bicubics defined on a uniform partition of the rectangular domain. The second algorithm allows for a general second order differential operator with respect to one of the coordinates and nonuniform partition in that coordinate direction. Bialecki and Fairweather [4] presented two matrix decomposition algorithms for solving the linear systems arising when orthogonal spline collocation is applied to more general separable problems than those considered herein. These algorithms implement higher degree, tensor-product piecewise polynomial spaces defined on nonuniform partitions, but they require $O(N^3)$ operations.

It should be mentioned that, for the piecewise Hermite bicubic Galerkin method, a diagonal matrix Λ and matrix Z satisfying equations similar to those in (1.8) have not been determined so far probably because the explicit piecewise Hermite cubic Galerkin approximation to the eigenvalue problem (1.9) is unknown. Nevertheless, Bank [2] developed a fast direct method, similar to that described by (1.4)–(1.6), for the solution of the linear system arising when the finite element Galerkin method with piecewise Hermite bicubics is used to solve (1.1) on a uniform partition of Ω. While Bank's method also relies heavily on the use of FFT routines to yield a $O(N^2 \log N)$ operation count, the linear system corresponding to (1.4) involves block diagonal matrices, with 2×2 diagonal blocks, in place of the diagonal matrices Λ and I. Bank also developed a fast method for the case in which C^0 quadratics are used instead of piecewise Hermite bicubics. Again the eigensystem is not known. Unfortunately the description of Bank's algorithms is imprecise and rather difficult to follow.

After the present paper was refereed, the work of Sun and Zamani [19] was brought to our attention by Roland Sweet. In their paper, Sun and Zamani consider piecewise Hermite bicubic orthogonal spline collocation for Poisson's equation and develop a matrix decomposition algorithm for the solution of the corresponding linear system (1.7). Their algorithm is based on the fact that the eigenvalues of the matrix $B^{-1}A$ are real and distinct [12] and hence there exists a real nonsingular matrix Q such that $B^{-1}A = Q\Lambda Q^{-1}$. Sun and Zamani's algorithm, whose cost is also proportional to $N^2 \log N$, is more complicated and less efficient than the algorithms presented in Section 3 of this paper. Our algorithms hinge on the existence of a real nonsingular matrix Z that satisfies both equations in (1.8). In particular, utilization of the second equation in (1.8) distinguishes our algorithms and makes them much more straightforward.

2. Piecewise Hermite cubic orthogonal spline collocation eigenvalue problem.

2.1. Eigenfunction derivation. Let N be a positive integer, and let $\{x^{(k)}\}_{k=0}^{N}$ be a uniform partition of the interval $[a,b]$, that is, $x^{(k)} = a + kh$, $k = 0, \ldots, N$, where the stepsize $h = (b-a)/N$. Let $\mathcal{M}_3$ be the space of piecewise Hermite cubics on $[0,1]$ defined by

$$\mathcal{M}_3 = \{v \in C^1[a,b] : v|_{[x^{(k)}, x^{(k+1)}]} \in P_3, k = 0, \ldots, N-1\},$$

and let

$$\mathring{\mathcal{M}}_3 = \{v \in \mathcal{M}_3 : v(a) = v(b) = 0\}, \tag{2.1}$$

where P_3 denotes the set of all polynomials of degree ≤ 3. Note that the dimension of $\mathring{\mathcal{M}}_3$ is $2N$. Let the set of Gauss points $\{\xi^{(m)}\}_{m=1}^{2N}$ on $[a,b]$ be defined by

$$\xi^{(2k+1)} = x^{(k+1/2)} - \frac{h}{2\sqrt{3}}, \qquad \xi^{(2k+2)} = x^{(k+1/2)} + \frac{h}{2\sqrt{3}}, \qquad k = 0, \ldots, N-1, \tag{2.2}$$

where $x^{(k+1/2)} = (x^{(k)} + x^{(k+1)})/2$.

Consider the following eigenvalue problem:

$$-U''(\xi^{(m)}) = \lambda U(\xi^{(m)}), \quad m = 1, \ldots, 2N, \qquad U \in \mathcal{M}_3, \tag{2.3a}$$

$$U(a) = U(b) = 0. \tag{2.3b}$$

We now prove the following theorem; cf. [11,12] where different, but equivalent, expressions for the eigenvalues and eigenfunctions are derived. We believe that our approach is easier to follow; moreover, it can be easily generalized to other boundary conditions and higher degree piecewise polynomials.

THEOREM 2.1. *The eigenvalue problem (2.3) has $2N$ distinct positive eigenvalues given by*

$$\lambda_j^{\pm} = 12\left(\frac{8+\eta_j \pm \mu_j}{7-\eta_j}\right)h^{-2}, \qquad j = 1,\ldots,N-1, \tag{2.4}$$

$$\lambda_0 = 36h^{-2}, \qquad \lambda_N = 12h^{-2}, \tag{2.5}$$

where

$$\eta_j = \cos\left(\frac{j\pi}{N}\right), \qquad \mu_j = \sqrt{43+40\eta_j-2\eta_j^2}. \tag{2.6}$$

For $x \in [x^{(k)}, x^{(k+1)}]$, $k = 0,\ldots,N-1$, the corresponding eigenfunctions are given by

$$\begin{aligned} U_j^{\pm}(x) \;=\; & c_j^{\pm}\Big\{\sqrt{1-\eta_j}(1+\eta_j)\Big[19-\eta_j\mp 2\mu_j+6(2+\eta_j\mp\mu_j)\rho_k(x)- \\ & 36(1-\eta_j)\rho_k^2(x)+8(4+5\eta_j\pm\mu_j)\rho_k^3(x)\Big]\cos\left(\frac{kj\pi}{N}\right)+ \\ & \sqrt{1+\eta_j}\Big[(1+\eta_j)(19-\eta_j\mp 2\mu_j)-6(1-\eta_j)(2+\eta_j\mp\mu_j)\rho_k(x)- \\ & 36(1-\eta_j^2)\rho_k^2(x)-8(1-\eta_j)(4+5\eta_j\pm\mu_j)\rho_k^3(x)\Big]\sin\left(\frac{kj\pi}{N}\right)\Big\}, \\ & j = 1,\ldots,N-1, \end{aligned} \tag{2.7}$$

$$U_0(x) = c_0\rho_k(x)[4\rho_k^2(x)-1], \qquad U_N(x) = c_N(-1)^{k-1}[4\rho_k^2(x)-1], \tag{2.8}$$

where η_j, μ_j are given by (2.6), $\rho_k(x) = (x-x^{(k+1/2)})/h$, and $c_j^{\pm}$ in (2.7) and c_0, c_N in (2.8) are arbitrary constants.

Proof. Let λ and U denote a real nonzero eigenvalue and the corresponding eigenfunction of (2.3), respectively, and for $x \in [x^{(k)}, x^{(k+1)}]$, $k = 0,\ldots,N-1$, let

$$U(x) = \alpha_k + \beta_k(x-x^{(k+1/2)}) + \gamma_k\frac{(x-x^{(k+1/2)})^2}{2} + \delta_k\frac{(x-x^{(k+1/2)})^3}{6}. \tag{2.9}$$

It is shown in [12] (on page 66, with $\theta = 1/(2\sqrt{3})$) that (2.3a) implies

$$\alpha_k = -\left(\frac{1}{\lambda}+\frac{h^2}{24}\right)\gamma_k, \qquad \beta_k = -\left(\frac{1}{\lambda}+\frac{h^2}{72}\right)\delta_k, \tag{2.10}$$

and

$$\begin{bmatrix} r & -s \\ 1 & -t \end{bmatrix} \begin{bmatrix} \gamma_{k+1} \\ \delta_{k+1} \end{bmatrix} = \begin{bmatrix} r & s \\ -1 & -t \end{bmatrix} \begin{bmatrix} \gamma_k \\ \delta_k \end{bmatrix}, \tag{2.11}$$

where

$$r = \frac{h^2}{12} - \frac{1}{\lambda}, \qquad s = \frac{h}{2}\left(\frac{h^2}{36} - \frac{1}{\lambda}\right), \qquad t = \frac{2}{h}\left(\frac{h^2}{9} - \frac{1}{\lambda}\right). \tag{2.12}$$

It is also easy to show that the boundary condition (2.3b) yields

$$s\delta_0 = r\gamma_0, \qquad s\delta_{N-1} = -r\gamma_{N-1}. \tag{2.13}$$

Since $rt - s > 0$, it follows from (2.11) that

$$\begin{bmatrix} \gamma_{k+1} \\ \delta_{k+1} \end{bmatrix} = \frac{1}{rt-s} \begin{bmatrix} rt+s & 2st \\ 2r & rt+s \end{bmatrix} \begin{bmatrix} \gamma_k \\ \delta_k \end{bmatrix}. \tag{2.14}$$

Assume that

$$rst < 0. \tag{2.15}$$

Then it is easy to verify that

$$\begin{bmatrix} \frac{-1}{st} & 0 \\ 0 & \frac{1}{\sqrt{-rst}} \end{bmatrix} \begin{bmatrix} rt+s & 2st \\ 2r & rt+s \end{bmatrix} \begin{bmatrix} -st & 0 \\ 0 & \sqrt{-rst} \end{bmatrix} \tag{2.16}$$

$$= \begin{bmatrix} rt+s & -2\sqrt{-rst} \\ 2\sqrt{-rst} & rt+s \end{bmatrix}.$$

Note that (2.15) implies $|rt+s| < |rt-s|$. Let θ be such that

$$\cos\theta = \frac{rt+s}{rt-s}, \qquad \theta \in (0, \pi). \tag{2.17}$$

Then it follows from (2.14), (2.16), and (2.17) that

$$\begin{bmatrix} \gamma_k \\ \delta_k \end{bmatrix} = \begin{bmatrix} -st & 0 \\ 0 & \sqrt{-rst} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^k \begin{bmatrix} \frac{-1}{st} & 0 \\ 0 & \frac{1}{\sqrt{-rst}} \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \delta_0 \end{bmatrix},$$

and hence

$$\begin{aligned} \gamma_k &= \gamma_0 \cos(k\theta) + \delta_0 \tfrac{st}{\sqrt{-rst}} \sin(k\theta), \\ \delta_k &= \delta_0 \cos(k\theta) - \gamma_0 \tfrac{\sqrt{-rst}}{st} \sin(k\theta), \end{aligned} \tag{2.18}$$

since

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^k = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix}.$$

Equations (2.13), (2.18) and (2.17) give

$$\cot[(N-1)\theta] = -\frac{rt+s}{2\sqrt{-rst}} = -\cot\theta,$$

which, in turn, yields

$$\theta = \frac{j\pi}{N}, \qquad j = 1, \ldots, N-1. \tag{2.19}$$

Substitute (2.19) into (2.17), set $\eta_j = \cos(j\pi/N)$, and use (2.12) to obtain the following quadratic equation in λh^2:

$$(7-\eta_j)(\lambda h^2)^2 - 24(8+\eta_j)\lambda h^2 + 432(1-\eta_j) = 0. \tag{2.20}$$

The solutions of (2.20) lead to (2.4).

Recall that the above derivation of the eigenvalues of (2.3) was based on the assumption that (2.15) holds. It can be easily verified that $\lambda_j^{\pm}$ given by (2.4) satisfy

$$0 < \frac{\lambda_j^- h^2}{12} < \frac{3}{4}, \qquad 1 < \frac{\lambda_j^+ h^2}{12} < 3. \tag{2.21}$$

Thus it follows from (2.12) and (2.21) that (2.15) is indeed satisfied. The corresponding eigenfunctions, given by (2.7), are obtained from (2.9), (2.10), (2.12), (2.13), (2.18).

Two additional eigenvalues (2.5) and their corresponding eigenfunctions (2.8) are obtained from (2.9)–(2.13) by first setting $s = 0$ in (2.12), and then $r = 0$. For instance, if $s = 0$, then the second equation in (2.12) implies $\lambda = \lambda_0$, where λ_0 is given by (2.5). Further, (2.11) and (2.13) yield $\gamma_k = 0$, $\delta_k = \delta_0$, $k = 0, \ldots, N-1$, and the formula for U_0 in (2.8) is obtained from (2.9), (2.10). It follows from (2.21) that all eigenvalues given by (2.4) and (2.5) are positive. Moreover, they are distinct since the first derivative of the function $(8 + x \pm \sqrt{43 + 40x - 2x^2})/(7-x)$ is never equal to zero. □

COROLLARY 2.1. *Let $U_j^{\pm}$, $j = 1, \ldots, N-1$, U_0, U_N be the eigenfunctions of (2.3) given by (2.7), (2.8), and let*

$$c_j^{\pm} = \frac{3}{2}\left[\frac{3}{2(b-a)}\right]^{1/2}(1+\eta_j)^{-1/2}\nu_j^{\pm}, \qquad j = 1, \ldots, N-1, \tag{2.22}$$

$$c_0 = 3\left[\frac{3}{2(b-a)}\right]^{1/2}, \qquad c_N = \frac{3}{2}\left[\frac{1}{2(b-a)}\right]^{1/2}, \tag{2.23}$$

where

$$\nu_j^{\pm} = \left[27(1+\eta_j)(8+\eta_j \mp \mu_j)^2 + (1-\eta_j)(11+7\eta_j \mp 4\mu_j)^2\right]^{-1/2}.$$

Then $U_j^{\pm}$, $j = 1, \ldots, N-1$, U_0, U_N *are orthonormal with respect to the inner product*

$$< u, v >= h \sum_{m=1}^{2N} u(\xi^{(m)}) v(\xi^{(m)}). \tag{2.24}$$

Proof. Since all of the eigenvalues are distinct, the orthogonality of $U_j^{\pm}$, $j = 1, \ldots, N-1$, U_0, U_N follows from Theorem 3.6 in [7], where it is shown that the second order derivative operator is self-adjoint with respect to (2.24) on $\mathring{\mathcal{M}}_3$. The normalization constants (2.22), (2.23) are obtained using the identities

$$\sum_{k=1}^{N-1} \sin^2\left(\frac{kj\pi}{N}\right) = \frac{N}{2}, \qquad \sum_{k=1}^{N-1} \sin\left(\frac{kj\pi}{N}\right)\cos\left(\frac{kj\pi}{N}\right) = 0, \qquad j = 1, \ldots, N-1.$$

□

The following corollary is used in Section 3, where we consider a specific basis for $\mathring{\mathcal{M}}_3$.

Corollary 2.2. *Let the assumptions of Corollary 2.1 be satisfied. Then, for* $j = 1, \ldots, N-1$,

$$U_j^{\pm}(x^{(k)}) = \gamma \alpha_j^{\pm} \sin\left(\frac{kj\pi}{N}\right), \qquad k = 0, \ldots, N,$$

$$\frac{d}{dx} U_j^{\pm}(x^{(k)}) = \gamma \beta_j^{\pm} h^{-1} \cos\left(\frac{kj\pi}{N}\right), \qquad k = 0, \ldots, N,$$

where

$$\gamma = 6\left[\frac{3}{2(b-a)}\right]^{1/2}, \qquad \alpha_j^{\pm} = (5+4\eta_j \mp \mu_j)\nu_j^{\pm}, \qquad \beta_j^{\pm} = 18\sin\left(\frac{j\pi}{N}\right)\nu_j^{\pm}. \tag{2.25}$$

Similarly, for $k = 0, \ldots, N$,

$$U_0(x^{(k)}) = U_N(x^{(k)}) = 0, \qquad \frac{d}{dx}U_0(x^{(k)}) = \gamma h^{-1}, \frac{d}{dx}U_N(x^{(k)}) = (-1)^k \frac{\gamma}{\sqrt{3}} h^{-1}.$$

2.2. Eigenvector derivation. Let $\{\phi_n\}_{n=1}^{2N}$ be a set of basis functions for $\mathring{\mathcal{M}}_3$, and let

$$U(x) = \sum_{n=1}^{2N} z_n \phi_n(x).$$

Then the eigenvalue problem (2.3) becomes the generalized eigenvalue problem

$$A\vec{z} = \lambda B\vec{z}, \tag{2.26}$$

where $\vec{z} = [z_1, \ldots, z_{2N}]^T$, and

$$A = (a_{mn})_{m,n=1}^{2N}, \quad a_{mn} = -\phi_n''(\xi^{(m)}), \qquad B = (b_{mn})_{m,n=1}^{2N}, \quad b_{mn} = \phi_n(\xi^{(m)}). \tag{2.27}$$

The following theorem gives the solution to (2.26).

THEOREM 2.2. *Let $\lambda_j^{\pm}$, $j = 1, \ldots, N-1$, λ_0, λ_N be the eigenvalues of (2.3) given by Theorem 2.1, and let $U_j^{\pm}$, $j = 1, \ldots, N-1$, U_0, U_N be the corresponding normalized eigenfunctions given by Corollary 2.1. Assume that*

$$U_j^{\pm}(x) = \sum_{n=1}^{2N} z_{n,j}^{\pm} \phi_n(x), \qquad U_0(x) = \sum_{n=1}^{2N} z_{n,0} \phi_n(x), \qquad U_N(x) = \sum_{n=1}^{2N} z_{n,N} \phi_n(x),$$

and let

$$\Lambda = \operatorname{diag}(\lambda_1^-, \ldots, \lambda_{N-1}^-, \lambda_0, \lambda_1^+, \ldots, \lambda_{N-1}^+, \lambda_N), \tag{2.28}$$

$$Z = h^{1/2} \left[\vec{z}_1^-, \ldots, \vec{z}_{N-1}^-, \vec{z}_0, \vec{z}_1^+, \ldots, \vec{z}_{N-1}^+, \vec{z}_N\right],$$

where $\vec{z}_j^{\pm} = [z_{1,j}^{\pm}, \ldots, z_{2N,j}^{\pm}]^T$, and $\vec{z}_0 = [z_{1,0}, \ldots, z_{2N,0}]^T$, $\vec{z}_N = [z_{1,N}, \ldots, z_{2N,N}]^T$. Then

$$AZ = BZ\Lambda, \qquad Z^T B^T B Z = I, \tag{2.29}$$

where I is the identity matrix of order $2N$.

Proof. The first equation in (2.29) follows directly from Theorem 2.1, whereas the second equation is a consequence of Corollary 2.1. □

We now give explicit formulas for the matrix Z for one special choice of the basis $\{\phi_n\}_{n=1}^{2N}$ of $\mathring{\mathcal{M}}_3$. The basis functions are defined in the following way: for $k = 0, \ldots, N$,

$$\phi_n(x^{(k)}) = \delta_{n,k}, \qquad \phi_n'(x^{(k)}) = 0, \qquad n = 1, \ldots, N-1, \tag{2.30a}$$

(2.30b) $\qquad \phi_{N+n}(x^{(k)}) = 0, \qquad \phi'_{N+n}(x^{(k)}) = h^{-1}\delta_{n,k}, \qquad n = 0, \ldots, N,$

where $\delta_{n,k}$ is the Kronecker delta. Note that this is the standard basis for $\mathring{\mathcal{M}}_3$ (see, for example, [14]), but the ordering of the basis functions is nonstandard. To describe Z, we require additional notation. Let S, C, and $\tilde{C}$ be matrices given by

$$S = \left(\sin\frac{kj\pi}{N}\right)_{k,j=1}^{N-1}, \qquad C = \left(\cos\frac{kj\pi}{N}\right)_{k,j=0}^{N},$$

and

$$\tilde{C} = \left(\cos\frac{kj\pi}{N}\right)_{k=0,j=1}^{N,N-1},$$

respectively, and let $\Lambda_\alpha^\pm$, $\Lambda_\beta^\pm$ be diagonal matrices defined by

$$\Lambda_\alpha^\pm = \mathrm{diag}(\alpha_1^\pm, \ldots, \alpha_{N-1}^\pm),$$

$$\Lambda_\beta^- = \mathrm{diag}(\beta_1^-, \ldots, \beta_{N-1}^-), \qquad \Lambda_\beta^+ = \mathrm{diag}(1, \beta_1^+, \ldots, \beta_{N-1}^+, 1/\sqrt{3}),$$

respectively, where $\alpha_j^\pm$, $\beta_j^\pm$ are given by (2.25). The following theorem follows easily from Theorem 2.2 and Corollary 2.2.

THEOREM 2.3. *Let Λ be as in (2.28), and let*

$$Z = 3\left(\frac{6}{N}\right)^{1/2}\left[\begin{array}{c|c|c|c} S\Lambda_\alpha^- & \vec{0} & S\Lambda_\alpha^+ & \vec{0} \\ \hline \tilde{C}\Lambda_\beta^- & \multicolumn{3}{c}{C\Lambda_\beta^+} \end{array}\right], \tag{2.31}$$

where $\vec{0}$ is the $(N-1)$-dimensional zero column vector. Then equations (2.29) are satisfied.

3. Algorithms for solving piecewise Hermite bicubic orthogonal spline collocation equations.

3.1. Piecewise Hermite bicubic orthogonal spline collocation scheme. Consider the homogeneous Dirichlet problem for Poisson's equation given by (1.1). For $i = 1, 2$, let N_i be a positive integer and $\{x_i^{(k)}\}_{k=0}^{N_i}$ a uniform partition of the interval $[a_i, b_i]$ with the stepsize $h_i = (b_i - a_i)/N_i$. Let $\mathcal{M}_3^{(i)}$ be the space of piecewise Hermite cubics on $[a_i, b_i]$ defined by (2.1), and let $\{\phi_n^{(i)}\}_{n=1}^{2N_i}$ be the set of basis functions for $\mathring{\mathcal{M}}_3^{(i)}$ as in (2.30) with $x^{(k)}$, h and N replaced by $x_i^{(k)}$, h_i and N_i, respectively. The piecewise Hermite bicubic orthogonal collocation approximation

$$U(x_1, x_2) = \sum_{n_1=1}^{2N_1}\sum_{n_2=1}^{2N_2} u_{n_1,n_2}\phi_{n_1}^{(1)}(x_1)\phi_{n_2}^{(2)}(x_2), \tag{3.1}$$

to the solution u of (1.1) is obtained by requiring that

$$-\Delta U(\xi_1^{(m_1)},\xi_2^{(m_2)}) = f(\xi_1^{(m_1)},\xi_2^{(m_2)}), \qquad m_i = 1,\ldots,2N_i, \qquad i = 1,2, \tag{3.2}$$

where $\{\xi_i^{(m)}\}_{m=1}^{2N_i}$ is the set of Gauss points on $[a_i,b_i]$ given by (2.2), with $x^{(k)}$, h and N replaced by $x_i^{(k)}$, h_i and N_i, respectively. Introducing

$$\vec{u} = [u_{1,1},\ldots,u_{1,2N_2},\ldots,u_{2N_1,1},\ldots,u_{2N_1,2N_2}]^T,$$

$$\vec{f} = [f_{1,1},\ldots,f_{1,2N_2},\ldots,f_{2N_1,1},\ldots,f_{2N_1,2N_2}]^T, \qquad f_{m_1,m_2} = f(\xi_1^{(m_1)},\xi_2^{(m_2)}),$$

equation (3.2) can be rewritten as the system of linear equations

$$(A_1 \otimes B_2 + B_1 \otimes A_2)\vec{u} = \vec{f}, \tag{3.3}$$

where the matrices A_i and B_i are defined in (2.27) with ϕ_n, $\xi^{(m)}$ and N replaced by $\phi_n^{(i)}$, $\xi_i^{(m)}$ and N_i, respectively. Because of the scaling factor h_i^{-1} in (2.30), all entries of A_i and B_i are $O(h_i^{-2})$ and $O(1)$, respectively.

3.2. Algorithm for solving system of linear equations. Let Λ_i and Z_i be defined by (2.28) and (2.31) with $[a,b]$, N and h replaced by $[a_i,b_i]$, N_i and h_i, respectively. It follows from Theorem 2.3 that, for $i = 1,2$,

$$Z_i^T B_i^T A_i Z_i = \Lambda_i, \qquad Z_i^T B_i^T B_i Z_i = I_i, \tag{3.4}$$

where I_i is the identity matrix of order $2N_i$. Using (3.4) and properties of the tensor product, it is easy to verify that

$$(Z_1^T B_1^T \otimes Z_2^T B_2^T)(A_1 \otimes B_2 + B_1 \otimes A_2)(Z_1 \otimes Z_2) = \Lambda_1 \otimes I_2 + I_1 \otimes \Lambda_2. \tag{3.5}$$

Equation (3.5) leads to the following matrix decomposition algorithm for solving (3.3).

ALGORITHM I

1. Compute $\vec{g} = (Z_1^T B_1^T \otimes Z_2^T B_2^T)\vec{f}$.
2. Solve $(\Lambda_1 \otimes I_2 + I_1 \otimes \Lambda_2)\vec{v} = \vec{g}$.
3. Compute $\vec{u} = (Z_1 \otimes Z_2)\vec{v}$.

Since there are at most four nonzero elements in each row of the matrices B_1 and B_2, matrix-vector multiplications involving the matrices B_1^T and B_2^T in step 1 require, in

total, $O(N_1N_2)$ arithmetic operations. It follows from (2.31) that FFT routines can be used to perform multiplications by the matrix Z_i^T in step 1, and Z_i in step 3. The corresponding cost of each of these steps is $O(N_1N_2 \log(N_1N_2))$. Step 2 involves the solution of a diagonal system of order $4N_1N_2$. Thus if $N_1 = N_2 = N$, the total cost of Algorithm I is $O(N^2 \log N)$.

Clearly, all steps of Algorithm I are highly suitable for parallel computations. For example, since

$$(Z_1^T B_1^T \otimes Z_2^T B_2^T) = (Z_1^T B_1^T \otimes I_2)(I_1 \otimes Z_2^T B_2^T),$$

the vector $\vec{g}$ in step 1 can be computed in parallel by first determining

$$(I_1 \otimes Z_2^T B_2^T)\vec{f} = [Z_2^T B_2^T \vec{f}_1, \ldots, Z_2^T B_2^T \vec{f}_M]^T \equiv \vec{f}^*,$$

where $\vec{f_m} = [f_{m,1}, \ldots, f_{m,M}]^T$, and then

$$(Z_1^T B_1^T \otimes I_2)\vec{f}^* = P^T(I_2 \otimes Z_1^T B_1^T)P\vec{f}^*,$$

where P is a permutation matrix such that

$$P\vec{f}^* = [f^*_{1,1}, \ldots, f^*_{M,1}, \ldots, f^*_{1,M}, \ldots, f^*_{M,M}]^T.$$

Similar remarks apply to the computation of $\vec{u}$ in step 3 since

$$(Z_1 \otimes Z_2) = (Z_1 \otimes I_2)(I_1 \otimes Z_2).$$

3.3. An extension. A matrix decomposition algorithm can easily be formulated for the solution of the homogeneous Dirichlet problem on a rectangle for the partial differential equation

$$-\frac{\partial^2 u}{\partial x_1^2} + cu + L_2 u = f(x_1, x_2) \tag{3.6a}$$

or

$$L_1 u - \frac{\partial^2 u}{\partial x_2^2} + cu = f(x_1, x_2), \tag{3.6b}$$

where c is a constant and

$$L_i u = -p(x_i)\frac{\partial^2 u}{\partial x_i^2} + q(x_i)\frac{\partial u}{\partial x_i} + r(x_i)u.$$

For the sake of argument, consider the problem involving (3.6a). Let $\{\phi_n^{(1)}\}_{n=1}^{2N_1}$ be the set of basis functions for $\overset{\circ}{\mathcal{M}}{}_3^{(1)}$ given in (2.30), and let $\{\phi_n^{(2)}\}_{n=1}^{2N_2}$ be a set of basis

functions for $\overset{\circ}{\mathcal{M}}{}_3^{(2)}$. In this case, the approximate solution (3.1) is obtained by solving the linear system

$$(\tilde{A}_1 \otimes B_2 + B_1 \otimes A_2)\vec{u} = \vec{f}, \tag{3.7}$$

where

$$\tilde{A}_1 = A_1 + cB_1, \qquad A_2 = (a_{mn}^{(2)})_{m,n=1}^{2N}, \qquad a_{mn}^{(2)} = L_2\phi_n^{(2)}(\xi_2^{(m)}),$$

and the matrices A_1, B_1 and B_2 are defined in (2.27). Using (3.4) with $i = 1$, it is easy to see that

$$(Z_1^T B_1^T \otimes I_2)(\tilde{A}_1 \otimes B_2 + B_1 \otimes A_2)(Z_1 \otimes I_2) = \tilde{\Lambda}_1 \otimes B_2 + I_1 \otimes A_2,$$

where $\tilde{\Lambda}_1 = \Lambda_1 + cI_1$. Therefore, the algorithm for solving (3.7) takes the following form:

ALGORITHM II

1. Compute $\vec{g} = (Z_1^T B_1^T \otimes I_2)\vec{f}$.
2. Solve $(\tilde{\Lambda}_1 \otimes B_2 + I_1 \otimes A_2)\vec{v} = \vec{g}$.
3. Compute $\vec{u} = (Z_1 \otimes I_2)\vec{v}$.

Note that steps 1 and 3 are similar to steps 1 and 3 of Algorithm I. To analyze step 2, we assume that the basis functions $\{\phi_n^{(2)}\}_{n=1}^{2N_2}$ are either B-splines, Hermite-type or monomial basis functions ordered in a standard way [1]. Then step 2 consists of solving $2N_1$ almost block diagonal linear systems of order $2N_2$, the coefficient matrix of the l^{th} system being of the form $A_2 + \lambda_l B_2$. These systems can be solved effectively using the package COLROW [8], [9], or one of the new packages described in [16], [17], depending on the choice of $\{\phi_n^{(2)}\}_{n=1}^{2N_2}$. With the use of such a package, step 2 requires $O(N_1N_2)$ operations. Thus, if $N_1 = N_2 = N$, the total cost of Algorithm II is $O(N^2 \log N)$ operations. When applied to Poisson's equation, the cost of this algorithm is asymptotically half of that of Algorithm I which requires twice as many FFTs. However it should be noted that in Algorithm II a nonuniform mesh in the x_2 direction may be employed. Since step 2 consists of solving independent linear systems, Algorithm II, like Algorithm I, is well suited for parallel computation.

4. Numerical examples. In this section, we present the results of numerical experiments conducted on a representative sample of boundary value problems listed

in Table 1. Computations were carried out with parallel implementations of both algorithms on four shared memory machines, an Alliant FX/8, a Sequent Symmetry S81, an Encore Multimax, and an IBM 3090-600J.

TABLE 1

Test problems for elliptic solver

(1)	$-u_{xx} - u_{yy} = f$
	$u(x,y) = (x^2 - x)(y^2 - y)e^{-10((x-0.5)^2+(y-0.3)^2)}$
(2)	$-u_{xx} - 4u_{yy} - 100u = f$
	$u(x,y) = (y^2 - y)(\cos(2\pi x) - 1)$
(3)	$-u_{xx} - u_{yy} - e^y u_y + \sin(y)u = f$
	$u(x,y) = e^{(x+y)}(x^2 - x)(y^2 - y)$
(4)	$-u_{xx} - \cos(y)u_{yy} + \sin(y)u_y - e^y u = f$
	$u(x,y) = (y^2 - y)(\cos(2\pi x) - 1)$

In order to demonstrate how well the algorithms utilize the parallel processing capabilities of the machines, we present in Figures 1-4 the speedups achieved with Algorithm II for $N = 50, 75$, and 100 on the ACRF machines and $N = 100, 250$, and 500 on the IBM 3090-600J; those for Algorithm I are similar. (Note that, for $N = 500$, there are 10^6 unknowns.) In all experiments involving Algorithm II, we chose Hermite-type basis functions of the form (2.30) with the standard ordering for $\{\phi_n^{(2)}\}_{n=1}^{2N_2}$, and used the package COLROW [8], [9] to solve the almost block diagonal systems arising in Step 2.

In Figures 5-8, we plot on a log-log scale the absolute errors in the approximations to u, $\dfrac{\partial u}{\partial x_1}$, $\dfrac{\partial u}{\partial x_2}$, and $\dfrac{\partial^2 u}{\partial x_1 \partial x_2}$ at the mesh points, for $h = 1/10, 1/20, 1/40, 1/80$ and $1/160$. These results demonstrate the fourth order accuracy of these approximations, and, in particular, the superconvergence of the derivative approximations at the mesh points. This superconvergence phenomenon, which occurs not only in all of the examples reported in this paper but also in others examined by the authors, has not

been reported previously in the literature.

Dyksen and Rice [13] discussed the importance of scaling for the Hermite bicubic collocation equations and the difficulty of determining reliable *a priori* scaling rules. As mentioned earlier, we used the natural scaling factor h^{-1} to scale the "slope" basis functions in (2.30); cf. [14]. This is equivalent to scaling every second column of the matrices A_2 and B_2 so that the entries of these matrices become $O(h_2^{-2})$ and $O(1)$, respectively. Without this scaling, a catastrophic loss of accuracy occurred for values of N as small as 50. COLROW, the package used in our tests, implements a variant of Gaussian elimination with partial pivoting, but not scaled partial pivoting as recommended by Dyksen and Rice.

5. Concluding remarks. Variants of Algorithms I and II which handle more general boundary conditions than (1.1b) may be developed systematically. Moreover, the algorithms can be extended to problems in higher dimensions in a straightforward fashion.

Since Algorithms I and II are based on the use of piecewise Hermite bicubics, they produce approximations of fourth order accuracy. The extension of these algorithms to higher order approximations using higher degree piecewise polynomials is under investigation. We are also exploring the use of the new algorithms as preconditioners in the solution of nonseparable problems, in domain decomposition methods for solving problems in more general regions, and in solution procedures for time dependent problems.

Acknowledgements. This research was supported in part by funds from the National Science Foundation Grant RII-8610671 and the Commonwealth of Kentucky through the University of Kentucky's Center for Computational Sciences. Computations were performed at the University of Kentucky, the Advanced Computing Research Facility, Mathematics and Computer Sciences Division, Argonne National Laboratory, and at the Cornell National Supercomputer Facility, a resource of the Center for Theory and Simulation in Science and Engineering, which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of the Corporate Research Institute.

REFERENCES

[1] U. Ascher, S. Pruess, and R. D. Russell, *On spline basis selection for solving differential equations*, SIAM J. Numer. Anal., 20 (1983), pp. 121–142.

[2] R. E. Bank, *Efficient algorithms for solving tensor product finite element equations*, Numer. Math., 31 (1978), pp. 49–61.

[3] B. Bialecki, *An alternating direction implicit method for orthogonal spline collocation linear systems*. Submitted.

[4] B. Bialecki and G. Fairweather, *Matrix decomposition algorithms in orthogonal spline collocation for separable elliptic boundary value problems*. Submitted.

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

[6] B. L. Buzbee, G. H. Golub, and C. W. Nielson, *On direct methods for solving Poisson's equations*, SIAM J. Numer. Anal., 7 (1970), pp. 627–656.

[7] K. D. Cooper and P. M. Prenter, *Alternating direction collocation for separable elliptic partial differential equations*. Submitted.

[8] J. C. Diaz, G. Fairweather, and P. Keast, *FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination*, ACM Trans. Math. Softw., 9 (1983), pp. 358–375.

[9] ———, *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. Softw., 9 (1983), pp. 376–380.

[10] F. W. Dorr, *The direct solution of the discrete Poisson equation on a rectangle*, SIAM Review, 12 (1970), pp. 248–263.

[11] W. R. Dyksen, *Tensor Product Generalized Alternating Direction Implicit Methods for Solving Separable Second Order Linear Elliptic Partial Differential Equations*, PhD thesis, Purdue University, West Lafayette, Indiana, 1982.

[12] ———, *Tensor product generalized ADI methods for separable elliptic problems*, SIAM J. Numer. Anal., 24 (1987), pp. 59–76.

[13] W. R. Dyksen and J. R. Rice, *The importance of scaling for the Hermite bicubic collocation equations*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 707–719.

[14] G. Fairweather, *Finite Element Galerkin Methods for Differential Equations*, Lecture Notes in Pure and Applied Mathematics, Vol. 34, Marcel Dekker, New York, 1978.

[15] R. E. Lynch, J. R. Rice, and D. H. Thomas, *Direct solution of partial difference equations by tensor product methods*, Numer. Math., 6 (1964), pp. 185–199.

[16] F. Majaess, P. Keast, and G. Fairweather, *The solution of almost block diagonal linear systems arising in spline collocation at Gaussian points with monomial basis functions*, ACM Trans. Math. Softw. To appear.

[17] F. Majaess, P. Keast, G. Fairweather, and K. R. Bennett, *Algorithm XXX : ABDPACK and ABBPACK - FORTRAN programs for the solution of almost block diagonal linear systems arising in spline collocation at Gaussian points with monomial basis functions*, ACM Trans. Math. Softw. To appear.

[18] J. R. Rice and R. F. Boisvert, *Solving Elliptic Problems Using ELLPACK*, Spring-Verlag, New York, 1985.

[19] W. Sun and N. G. Zamani, *A fast algorithm for solving the tensor product collocation equations*, J. Franklin Inst., 326 (1989), pp. 295–307.

[20] P. N. Swarztrauber, *Fast Poisson solvers*, in Studies in Numerical Analysis, G. H. Golub, ed., vol. 24 of MAA Studies in Mathematics, Mathematical Association of America, 1984, pp. 319–370.

[21] P. N. Swarztrauber and R. A. Sweet, *Vector and parallel methods for the direct solution of Poisson's equation*, J. Comp. Appl. Math., 27 (1989), pp. 241–263.

[22] E. L. Wachspress, *Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics*, Prentice Hall, Englewood Cliffs, New Jersey, 1966.

FIG. 1. *Speedups on the Alliant FX/8. Algorithm I.*

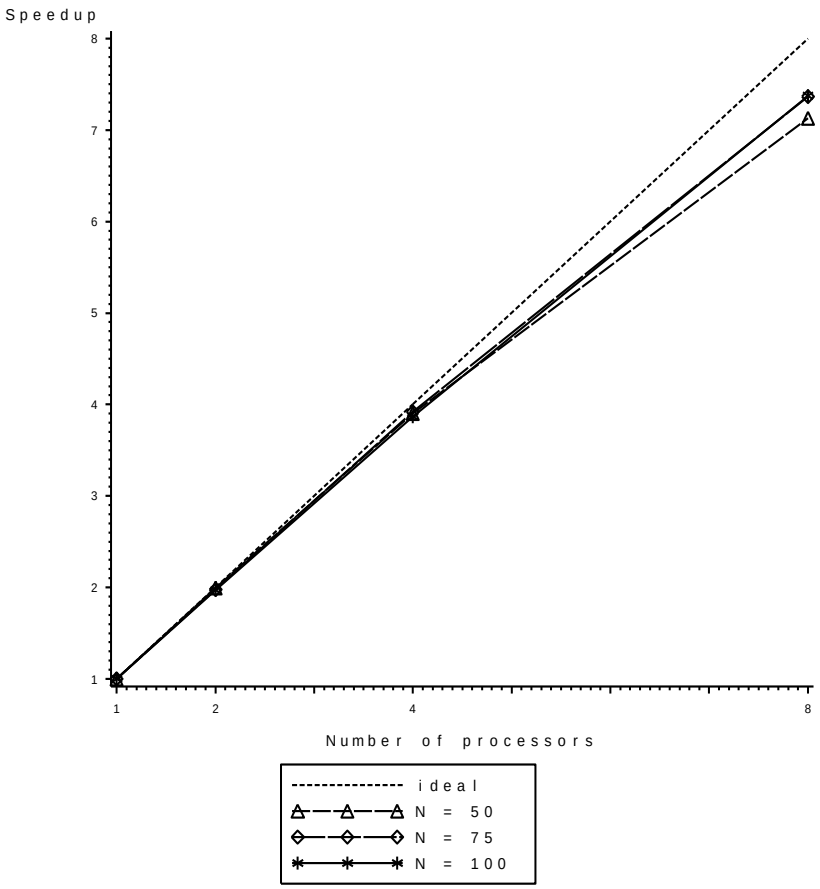

FIG. 2. *Speedups on the Sequent Symmetry S81. Algorithm I.*

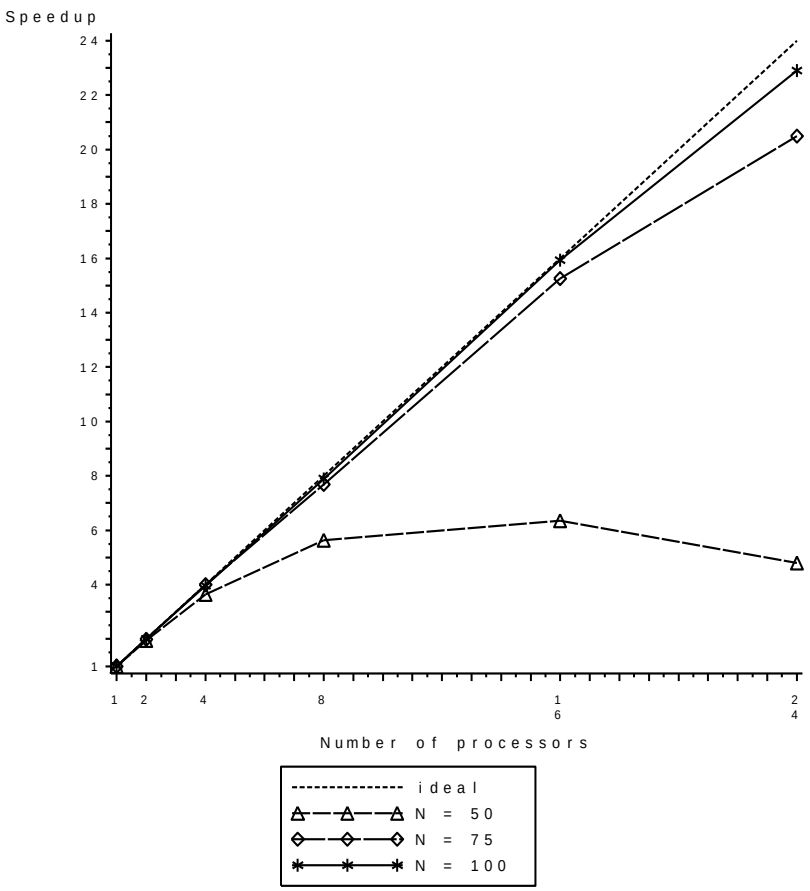

FIG. 3. *Speedups on the Encore Multimax. Algorithm I.*

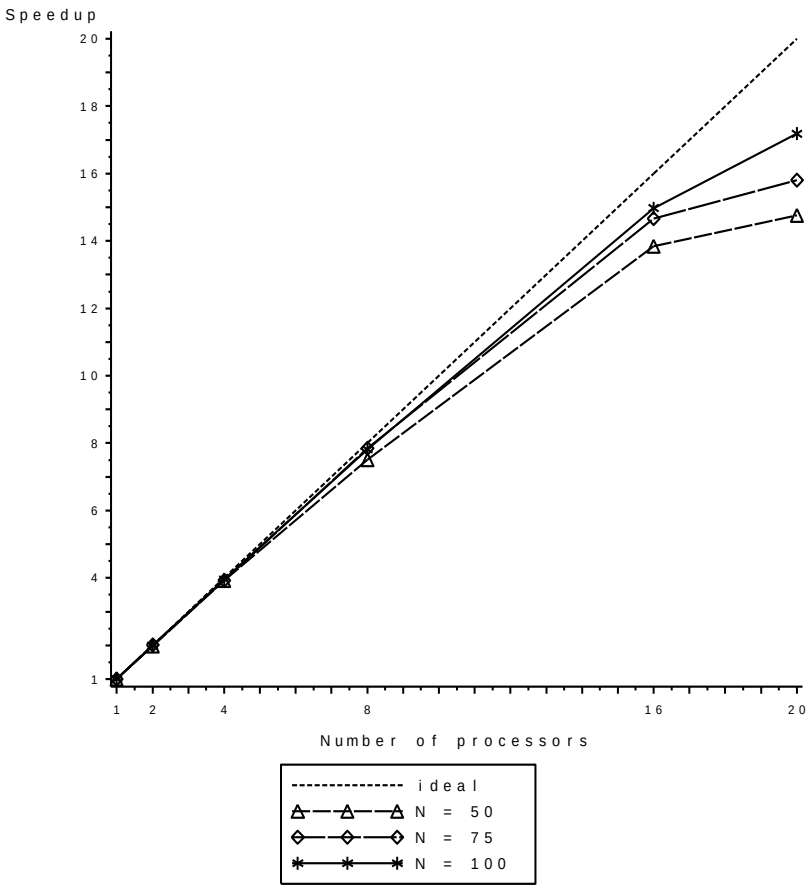

FIG. 4. *Speedups on the IBM 3090 600J. Algorithm I.*

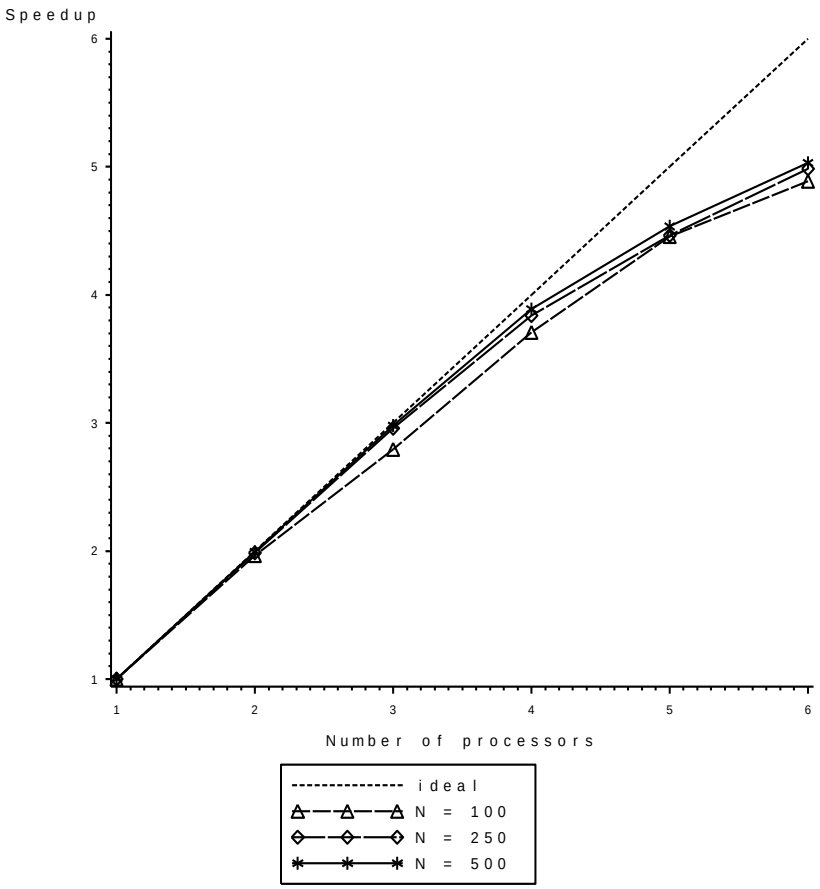

FIG. 5. *Convergence results for Problem 1. Algorithm II.*

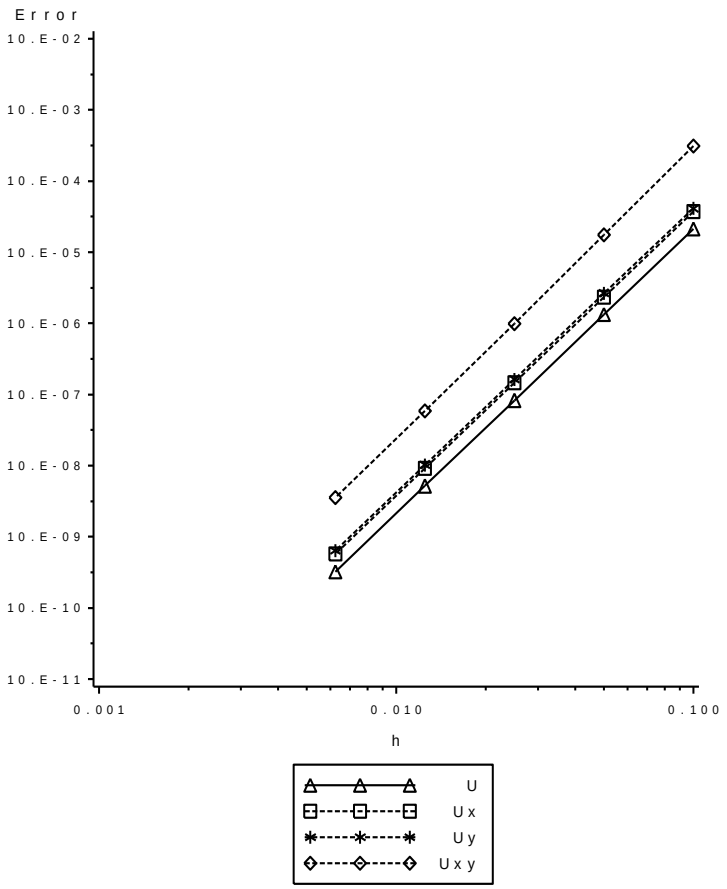

FIG. 6. *Convergence results for Problem 2. Algorithm II.*

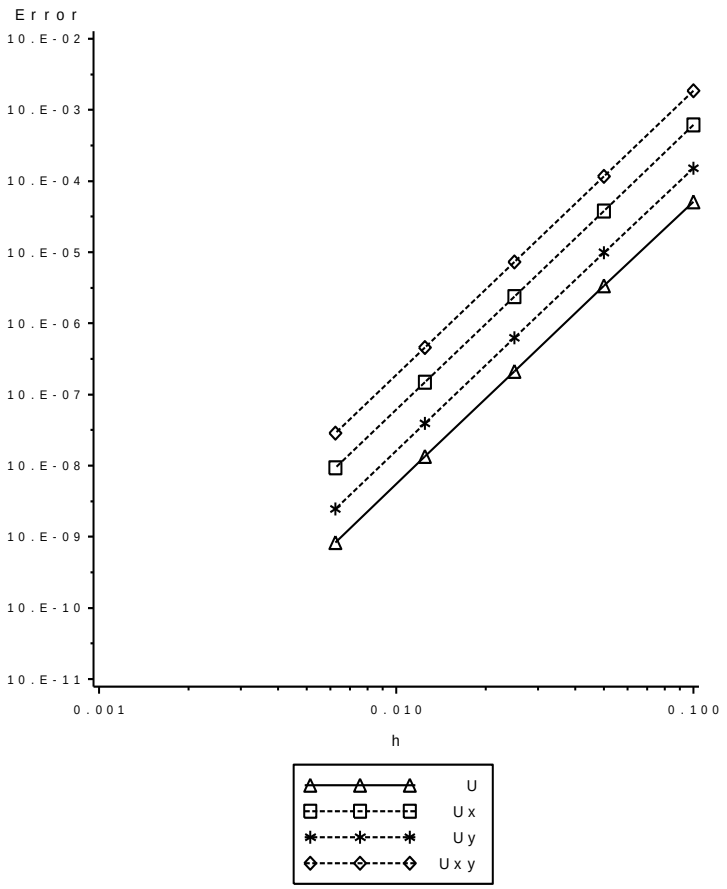

FIG. 7. *Convergence results for Problem 3. Algorithm II.*

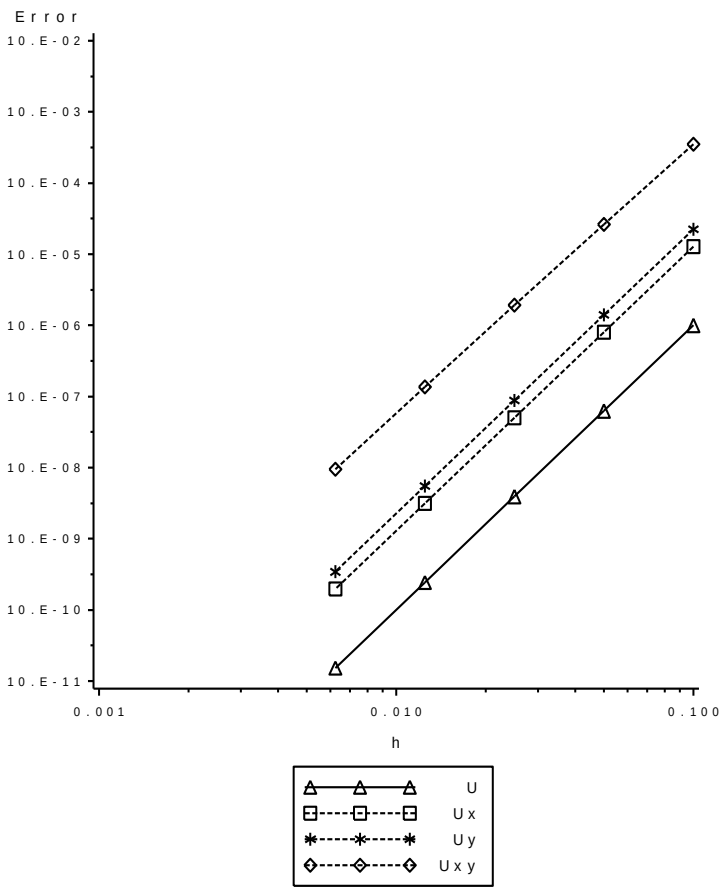
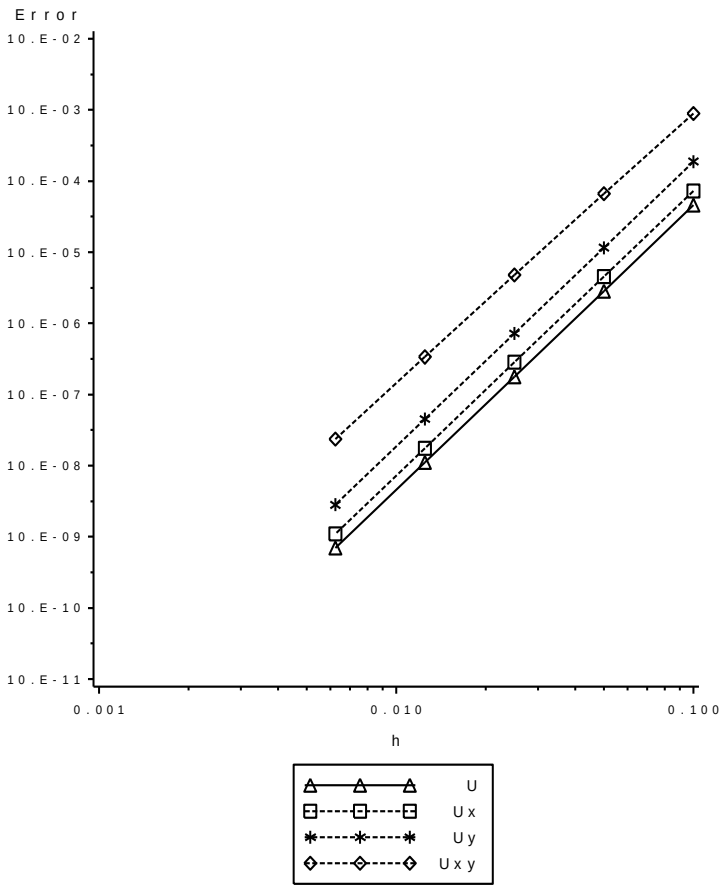
FIG. 8. *Convergence results for Problem 4. Algorithm II.*