View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Journal of Computational and Applied Mathematics 33 (1990) 141–155
North-Holland

On efficient direct methods for conforming spectral domain decomposition techniques

Andreas KARAGEORGHIS
Mathematics Department, Southern Methodist University, Dallas, TX 75275-0156, USA

Timothy N. PHILLIPS
Department of Mathematics, University College of Wales, Aberystwyth, UK SY23 3BZ

Received 22 December 1989
Revised 23 April 1990

Abstract: A conforming spectral domain decomposition technique is described for the solution of Stokes flow in rectangularly decomposable domains. The matrices arising from such a spectral discretization procedure possess a block tridiagonal structure where these blocks are full submatrices. Efficient direct solution procedures are proposed to take advantage of the matrix structure. A comparison of the methods in terms of computational efficiency is made. Numerical results are presented for the flow through an abruptly contracting channel.

Keywords: Spectral methods, collocation, domain decomposition.

1. Introduction

The aim of this paper is to compare several direct methods for solving the algebraic equations which result from spectral discretizations of the biharmonic equation in rectangularly decomposable domains using domain decomposition techniques. Unlike their finite difference and finite element counterparts these systems of equations are not sparse. However, they do possess a block tridiagonal structure, with zero entries elsewhere. We seek efficient ways of inverting such systems exploiting this fact. Although the techniques in this paper are described for the solution of Stokes flow they can be easily generalized to solve flows with inertia.

Efficient techniques for the inversion of matrices associated with spectral domain decomposition methods have been the subject of papers by Patera [9], who uses a direct fast solver for spectral element discretizations of second order separable elliptic equations, and Macaraeg and Streett [7], who adapt an influence matrix technique to spectral patching methods. In a recent paper, Phillips and Karageorghis [10] describe an application of the capacitance matrix technique [2] to the system resulting from a nonconforming spectral collocation discretization of flow through an abruptly contracting channel. In the same paper a coefficient splitting technique which partitions the matrix in such a way that the size of the blocks correspond to the number of unknown expansion coefficients in each subdomain is described.

0377-0427/90/$03.50 © 1990 – Elsevier Science Publishers B.V. (North-Holland)

The present study compares these two techniques and also a NAG library subroutine which solves almost block diagonal systems [1] on the standard contraction flow problem described in detail in Section 2. This latter technique and the capacitance matrix method are comparable in terms of computational time and the number of storage locations, and both outperform the coefficient splitting method. However, for large numbers of degrees of freedom the capacitance matrix method suffers from poor conditioning as a result of almost singular submatrices. This is avoided in the NAG subroutine for almost block diagonal systems by the use of global pivoting. Therefore we are able to use more degrees of freedom when using this technique and hence obtain additional accu acy.

2. The governing equations

The governing equations for the planar inertialess flow of an incompressible Newtonian fluid assume the mathematical form

$$\nabla \cdot \boldsymbol{v} = 0, \tag{2.1}$$

$$\nabla \cdot \boldsymbol{\sigma} = 0, \tag{2.2}$$

where $\boldsymbol{v} = (u, v)$ denotes the velocity field and $\boldsymbol{\sigma}$ the Cauchy stress tensor. These statements are the conservation of mass and momentum, respectively. For a Newtonian fluid, the extra stress tensor T and rate of deformation tensor D are related by

$$T = 2\eta D, \tag{2.3}$$

where η is a material constant and $D = \frac{1}{2}[\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^{\mathrm{T}}]$. For an incompressible fluid, the motion of the continuum determines the stress tensor up to an arbitrary isotropic tensor and thus $\boldsymbol{\sigma}$ and T are related as follows

$$\boldsymbol{\sigma} = -pI + T, \tag{2.4}$$

where p is an arbitrary pressure and I is the identity tensor.

If we define a stream function ψ by

$$u = \frac{\partial \psi}{\partial y}, \qquad v = -\frac{\partial \psi}{\partial x},$$

then (2.1) is satisfied identically. Substitution of T from (2.3) into (2.4) and then substitution of $\boldsymbol{\sigma}$ from (2.4) into (2.2) results in the equation

$$-\nabla p + 2\eta \nabla \cdot D = 0. \tag{2.5}$$

The pressure may be eliminated by taking the curl of (2.5) to give a biharmonic equation for the stream function

$$\nabla^4 \psi = 0, \tag{2.6}$$

since $\mathrm{curl}(\nabla p) = 0$ and $\mathrm{curl}(\nabla \cdot D) = \nabla^4 \psi$.

We consider Stokes flow through a $1:\alpha$ contraction shown in Fig. 1. The stream function formulation has the advantage that the continuity equation is automatically satisfied and that no boundary conditions for the vorticity need to be manufactured.

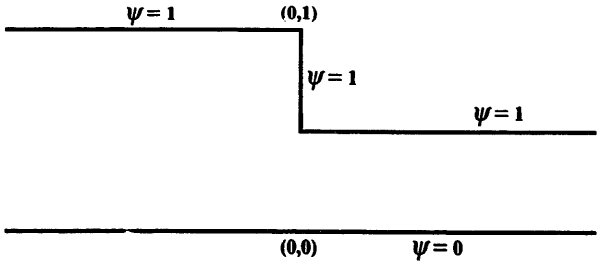

Fig. 1. Contraction geometry.

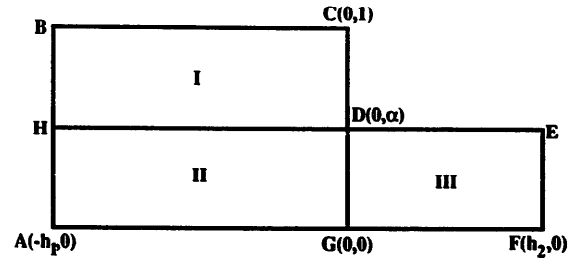

Fig. 2. Three element subdivision of domain.

Some of the boundary conditions are shown in Fig. 1. Since the flow is symmetric about the line $y=0$, only the upper half of the channel need be considered. No-slip constraints are imposed on the upper channel wall. Along $y=0$, the symmetry conditions

$$\psi=0, \qquad \frac{\partial^2\psi}{\partial y^2}=0 \tag{2.7}$$

are imposed. On entry and exit Poiseuille flow is assumed, defined by

$$\begin{aligned} &\psi(x,\ y)\to G(y) \quad \text{as } x\to-\infty, \quad 0\leqslant y\leqslant 1,\\ &\psi(x,\ y)\to G\left(\frac{y}{\alpha}\right) \quad \text{as } x\to+\infty, \quad 0\leqslant y\leqslant \alpha, \end{aligned} \tag{2.8}$$

where $G(y)=\frac{1}{2}y(3-y^2)$ (for details see [10]).

3. Conforming spectral collocation strategy

The flow region is truncated on entry and exit at finite distances h_1 and h_2 from the origin, respectively. The domain truncation means that fictitious boundary conditions need to be imposed on entry and exit; namely: $\psi(-h_1,\ y)=G(y)$ and $(\partial\psi/\partial x)(-h_1,\ y)=0$, $0\leqslant y\leqslant 1$ on entry; and $\psi(h_2,\ y)=G(y/\alpha)$ and $(\partial\psi/\partial x)(h_2,\ y)=0$, $0\leqslant y\leqslant\alpha$ on exit. The distances h_1, h_2 at which the domain is truncated need to be sufficiently large so that the flow is fully developed in the entry and exit sections.

Following [11], the truncated domain is subsequently divided into three elements, as shown in Fig. 2. Three elements are needed in order to achieve conforming approximations. This is in contrast to the two-element formulation of [6].

In each element the streamfunction $\psi(x,\ y)$ is approximated by $\psi^k(x,\ y)$, where

$$\psi^k(x,\ y)=G^k(y)+\sum_{n=N_0^k}^{N^k}\ \sum_{m=M_0^k}^{M^k} a_{mn}^k P_n^k(y)W_m^k(x), \quad k=\mathrm{I},\ \mathrm{II},\ \mathrm{III}, \tag{3.1}$$

and

$$G^{\mathrm{I}}(y)=G^{\mathrm{II}}(y)=G(y), \qquad G^{\mathrm{III}}(y)=G\left(\frac{y}{\alpha}\right),$$

$$N_0^{\mathrm{I}}=M_0^{\mathrm{I}}=N_0^{\mathrm{II}}=M_0^{\mathrm{II}}=M_0^{\mathrm{III}}=2, \qquad N_0^{\mathrm{III}}=4.$$

The polynomials $\{P_n^k(y)\}$, $\{W_m^k(x)\}$ are modified shifted Chebyshev polynomials which satisfy automatically all boundary conditions with the exception of the boundary conditions on the vertical wall CD. For example,

$$W_m^{\mathrm{I}}(x) = T_m^{\mathrm{I}}(x) + \alpha_m^{\mathrm{I}} T_1^{\mathrm{I}}(x) + \beta_m^{\mathrm{I}} T_0^{\mathrm{I}}(x), \quad 2 \leqslant m \leqslant M^{\mathrm{I}}, \tag{3.2}$$

where $T_m^{\mathrm{I}}(x)$, $0 \leqslant m \leqslant M^{\mathrm{I}}$, are shifted Chebyshev polynomials on $[-h_1, 0]$ defined by

$$T_m^{\mathrm{I}}(x) = T_m\left(\frac{2x + h_1}{h_1}\right),$$

and α_m^{I}, β_m^{I} are given by

$$\alpha_m^{\mathrm{I}} = (-1)^m m^2, \quad \beta_m^{\mathrm{I}} = (-1)^m (m^2 - 1), \qquad 2 \leqslant m \leqslant M^{\mathrm{I}}.$$

Similarly, we can show that

$$P_n^{\mathrm{I}}(y) = \tilde{T}_n^{\mathrm{I}}(y) + \tilde{\alpha}_n^{\mathrm{I}} \tilde{T}_1^{\mathrm{I}}(y) + \tilde{\beta}_n^{\mathrm{I}} \tilde{T}_0^{\mathrm{I}}(y), \quad 2 \leqslant n \leqslant N^{\mathrm{I}}, \tag{3.3}$$

where $\tilde{T}_n^{\mathrm{I}}(y)$, $0 \leqslant n \leqslant N^{\mathrm{I}}$, are the shifted Chebyshev polynomials on $[\alpha, 1]$ defined by

$$\tilde{T}_n^{\mathrm{I}}(y) = T_n\left(\frac{2y - 1 - \alpha}{1 - \alpha}\right),$$

and $\tilde{\alpha}_n^{\mathrm{I}}$, $\tilde{\beta}_n^{\mathrm{I}}$ are given by

$$\tilde{\alpha}_n^{\mathrm{I}} = -n^2 - 1, \qquad \tilde{\beta}_n^{\mathrm{I}} = n^2 - 1.$$

In approximation ψ^{III}, we take $N_0^{\mathrm{III}} = 4$, because the corresponding modified Chebyshev polynomials in the y-direction are chosen to satisfy the boundary conditions on both DE and GF automatically. Further, for the approximation to be conforming, we require $M^{\mathrm{I}} = M^{\mathrm{II}}$ and $N^{\mathrm{II}} = N^{\mathrm{III}}$.

The coefficients $\{a_{mn}^k\}$ in the expansions (3.1) are determined by collocating the governing equation at certain points in each element. These points are chosen from a subset of the points where the Chebyshev polynomials of highest degree used in (3.1) reach their extrema. In region I, for example, these points are given by

$$x_i^{\mathrm{I}} = \frac{h_1(x_i - 1)}{2}, \qquad y_j^{\mathrm{I}} = \frac{(1-\alpha) y_j + 1 + \alpha}{2},$$

where

$$x_i = -\cos\left(\frac{i\pi}{M_{\mathrm{I}}}\right), \quad 0 \leqslant i \leqslant M^{\mathrm{I}},$$

$$y_j = -\cos\left(\frac{j\pi}{N_{\mathrm{I}}}\right), \quad 0 \leqslant j \leqslant N^{\mathrm{I}}.$$

The boundary conditions along CD are satisfied at all points on that segment by collocating the boundary conditions at enough points. Finally, across the element interfaces we impose continuity of ψ and the normal derivatives $\partial\psi/\partial n$, $\partial^2\psi/\partial n^2$ and $\partial^3\psi/\partial n^3$. On the interface HD this is done by imposing

$$\frac{\partial^k \psi^{\mathrm{I}}}{\partial y^k}(x, \alpha) = \frac{\partial^k \psi^{\mathrm{II}}}{\partial y^k}(x, \alpha), \quad k = 0, 1, 2, 3, \quad \text{at the points } x_i^{\mathrm{I}}, \quad i = 2, 3, \ldots, M^{\mathrm{I}} - 2. \tag{3.4}$$

The extra boundary conditions

$$\psi^{\mathrm{II}}(0,\alpha)=1,\qquad \frac{\partial\psi^{\mathrm{II}}}{\partial x}(0,\alpha)=0,\qquad \frac{\partial\psi^{\mathrm{II}}}{\partial y}(0,\alpha)=0,\qquad \frac{\partial^2\psi^{\mathrm{II}}}{\partial x\,\partial y}(0,\alpha)=0, \tag{3.5}$$

ensure pointwise continuity of ψ and $\partial\psi/\partial y$ across HD. The continuity of the second and third derivatives in (3.4) is, on the other hand, only imposed at $M^{\mathrm{I}}-4$ points. This is so because pointwise continuity of these derivatives across HD would be inconsistent due to their singular behaviour about the corner D. The interface DG is treated in a similar way (for details, see [11]).

4. Direct methods of solution

The global system for the expansion coefficients is

$$Ga=r, \tag{4.1}$$

where

$$G=\left[\begin{array}{c|c|c} A & 0 & 0 \\ \hline & B & 0 \\ \hline 0 & C & 0 \\ \hline 0 & D & \\ \hline 0 & 0 & E \end{array}\right],\qquad a=\left[\begin{array}{c} a^{\mathrm{I}} \\ \hline a^{\mathrm{II}} \\ \hline a^{\mathrm{III}} \end{array}\right],\qquad r=\left[\begin{array}{c} r_1 \\ \hline r_2 \\ \hline r_3 \\ \hline r_4 \\ \hline r_5 \end{array}\right]. \tag{4.2}$$

The dimension of the matrix G is $n_1+2n_2+n_3+2n_4+n_5$ where

$$n_1=(N^{\mathrm{I}}-3)(M^{\mathrm{I}}-3)+2(N^{\mathrm{I}}-1),\qquad n_2=2(M^{\mathrm{I}}-3),$$
$$n_3=(N^{\mathrm{II}}-3)(M^{\mathrm{II}}-3)+4,\qquad n_4=2(N^{\mathrm{II}}-3),\qquad n_5=(N^{\mathrm{III}}-3)(M^{\mathrm{III}}-3).$$

The matrix G is a 5×5 block tridiagonal matrix when G is partitioned to have square diagonal blocks of orders n_1, $2n_2$, n_3, $2n_4$ and n_5, respectively.

The vectors a^{I}, a^{II} and a^{III} are vectors containing the unknown coefficients in the expansions in regions I, II and III respectively.

The matrix A is of dimension $n_1(n_1+n_2)$. Its first $2(N^{\mathrm{I}}-1)$ rows correspond to the satisfaction of the boundary conditions along the vertical wall and the remaining rows correspond to the satisfaction of the governing equation at $(N^{\mathrm{I}}-3)(M^{\mathrm{I}}-3)$ points in region I. The first $2(N^{\mathrm{I}}-1)$ entries of the vector r_1 correspond to the boundary conditions along the vertical wall and the remaining entries correspond to the right-hand side of the governing equation in region I (in the case of the biharmonic equation all entries are equal to zero).

The matrix B is of dimension $2n_2(n_1+2n_2+n_3+n_4)$ and its rows correspond to the interface continuity conditions between regions I and II. In this case the vector r_2 is a zero vector of dimension $2n_2$.

The matrix C is of dimension $n_3(n_2+n_3+n_4)$. Its first four rows correspond to the extra boundary conditions imposed at the re-entrant corner (see [11]). The remaining rows correspond to the satisfaction of the governing equation at $(N^{\mathrm{II}}-3)(M^{\mathrm{II}}-3)$ collocation points in region II. The first four entries in the vector r_3 correspond to the boundary conditions at the re-entrant corner and the remaining entries to the right-hand sides of the governing equation (in this case all the entries are equal to zero).

The matrix D is of dimension $2n_4(n_2+n_3+2n_4+n_5)$ and its rows correspond to the interface continuity conditions between regions II and III. The vector r_4 is of dimension $2n_4$ and corresponds to the difference between $G^{\mathrm{III}}(y)$ and $G^{\mathrm{II}}(y)$ at the interface collocation points.

Finally, the matrix E is of dimension $n_5(n_4+n_5)$ and its rows correspond to the satisfaction of the governing equation at $(N^{\mathrm{III}}-3)(M^{\mathrm{III}}-3)$ collocation points in region III. The vector r_5 contains the right-hand sides of the governing equation (in this case all the entries are equal to zero).

The global matrix for the Navier–Stokes equations for the same problem is of exactly the same type as the matrix G. In the system (4.2) and in the description of each subsequent technique we assume the right-hand sides to be nonzero which is the case in the solution of the Navier–Stokes equations.

4.1. Capacitance matrix method

The global system (4.1) may be rewritten in the partitioned form

$$\left[\begin{array}{c|c|c|c|c} A_1 & A_2 & 0 & 0 & 0 \\ \hline B_1 & B_2 & B_3 & B_4 & 0 \\ \hline 0 & C_2 & C_3 & C_4 & 0 \\ \hline 0 & D_2 & D_3 & D_4 & D_5 \\ \hline 0 & 0 & 0 & E_4 & E_5 \end{array}\right] \left[\begin{array}{c} x_1 \\ \hline x_2 \\ \hline x_3 \\ \hline x_4 \\ \hline x_5 \end{array}\right] = \left[\begin{array}{c} r_1 \\ \hline r_2 \\ \hline r_3 \\ \hline r_4 \\ \hline r_5 \end{array}\right], \tag{4.3}$$

where A_1, B_2, C_3, D_4 and E_5 are square matrices of order n_1, $2n_2$, n_3, $2n_4$ and n_5, respectively. The basic idea is to rewrite system (4.3) into the natural component form suggested by the partitioning and eliminate the vectors x_1, x_3 and x_5. This gives rise to a much smaller system which may be solved to yield x_2 and x_4. Back substitution subsequently yields x_1, x_3 and x_5. In component form, system (4.3) gives

$$A_1x_1 + A_2x_2 = r_1, \tag{4.3.1}$$

$$B_1x_1 + B_2x_2 + B_3x_3 + B_4x_4 = r_2, \tag{4.3.2}$$

$$C_2x_2 + C_3x_3 + C_4x_4 = r_3, \tag{4.3.3}$$

$$D_2x_2 + D_3x_3 + D_4x_4 + D_5x_5 = r_4, \tag{4.3.4}$$

$$E_4x_4 + E_5x_5 = r_5. \tag{4.3.5}$$

We write x_1, x_3 and x_5 in terms of x_2 and x_4, by premultiplying (4.3.1), (4.3.3) and (4.3.5) by A_1^{-1}, C_3^{-1} and E_5^{-1} respectively:

$$x_1 = A_1^{-1}r_1 - A_1^{-1}A_2x_2, \tag{4.3.6}$$

$$x_3 = C_3^{-1}r_3 - C_3^{-1}C_2x_2 - C_3^{-1}C_4x_4, \tag{4.3.7}$$

$$x_5 = E_5^{-1}r_5 - E_5^{-1}E_4x_4. \tag{4.3.8}$$

Eliminating x_1, x_3 and x_5 from (4.3.2) and (4.3.4), we obtain a system of the form

$$\begin{cases} \tilde{B}_2x_2 + \tilde{B}_4x_4 = \tilde{r}_2 \\ \tilde{D}_2x_2 + \tilde{D}_4x_4 = \tilde{r}_4 \end{cases} \quad \text{or} \quad \tilde{F}\begin{bmatrix} \tilde{x}_2 \\ \tilde{x}_4 \end{bmatrix} = \begin{bmatrix} \tilde{r}_2 \\ \tilde{r}_4 \end{bmatrix}, \tag{4.3.9, 4.3.10}$$

where the matrices $\tilde{B}_2$, $\tilde{B}_4$, $\tilde{D}_2$ and $\tilde{D}_4$ are of the same size as B_2, B_4, D_2 and D_4, respectively.

The solution of (4.3.9), (4.3.10) yields x_2 and x_4. The remaining coefficients x_1, x_3 and x_5 may be obtained from (4.3.6)–(4.3.8).

In the above calculations advantage is taken of the fact that the matrices A_2, B_4, C_2, C_4, D_2 and E_4 are only half full, i.e., only half of their columns are nonzero and they are of the form

$$A_2 = [A_2^* \,|\, 0], \qquad B_4 = [B_4^* \,|\, 0], \qquad C_2 = [0 \,|\, C_2^*],$$
$$C_4 = [C_4^* \,|\, 0], \qquad D_2 = [0 \,|\, D_2^*], \qquad E_4 = [0 \,|\, E_4^*].$$

Advantage of the fact that A_2 is only half full is taken in the calculation of $A_1^{-1}r_1$ and $A_1^{-1}A_4$ by solving $A_1[W_1 \,|\, v_1] = [A_2^* \,|\, r_1]$, i.e., a system with only $n_2 + 1$ right-hand sides rather than $2n_2 + 1$ right-hand sides. Similarly, advantage of the structures of the matrices C_2, C_4 and E_4 is taken when calculating $C_3^{-1}C_2$, $C_3^{-1}C_4$ and $E_5^{-1}r_4$, respectively. Further, the presence of zero columns in B_4 and D_4 is exploited in the evaluation of $\tilde{B}_4$, $\tilde{B}_2$, $\tilde{D}_4$, $\tilde{D}_2$.

4.2. Coefficient splitting technique

The main feature in the coefficient splitting technique is the partition of the global matrix G in such a way so that the diagonal blocks correspond to the vector of unknown coefficients in the approximations to the stream function in each of the three elements. This natural splitting makes the implementation of the algorithm much easier than the implementation of the capacitance matrix technique. The global system (4.1) may be rewritten in the form

$$\left[\begin{array}{c|c|c} P & \begin{array}{c} 0 \\ \hline Q_2 \end{array} & 0 \\ \hline \begin{array}{c} Q_1 \\ \hline 0 \end{array} & R & \begin{array}{c} 0 \\ \hline S_2 \end{array} \\ \hline 0 & \begin{array}{c} S_1 \\ \hline 0 \end{array} & T \end{array}\right] \left[\begin{array}{c} a^{\mathrm{I}} \\ \hline a^{\mathrm{II}} \\ \hline a^{\mathrm{III}} \end{array}\right] = \left[\begin{array}{c} t_1 \\ \hline t_2 \\ \hline t_3 \end{array}\right]. \tag{4.4}$$

The square matrices P, R, T are of order $n_1 + n_2$, $n_2 + n_3 + n_4$ and $n_4 + n_5$, respectively and are related to the submatrices appearing in (4.2) in the following way:

$$\begin{array}{lll}
P(i, j) = A(i, j), & 1 \leqslant i \leqslant n_1, & 1 \leqslant j \leqslant n_1 + n_2, \\
P(n_1 + i, j) = B(i, j), & 1 \leqslant i \leqslant n_2, & 1 \leqslant j \leqslant n_1 + n_2, \\
R(i, j) = B(n_2 + i, n_1 + n_2 + j), & 1 \leqslant i \leqslant n_2, & 1 \leqslant j \leqslant n_2 + n_3 + n_4, \\
R(n_2 + i, j) = C(i, j), & 1 \leqslant i \leqslant n_3, & 1 \leqslant j \leqslant n_2 + n_3 + n_4, \\
R(n_2 + n_3 + i, j) = D(i, j), & 1 \leqslant i \leqslant n_4, & 1 \leqslant j \leqslant n_2 + n_3 + n_4, \\
T(i, j) = D(n_4 + i, n_2 + n_3 + n_4 + j), & 1 \leqslant i \leqslant n_4, & 1 \leqslant j \leqslant n_4 + n_5, \\
T(n_4 + i, j) = E(i, j), & 1 \leqslant i \leqslant n_5, & 1 \leqslant j \leqslant n_4 + n_5.
\end{array}$$

The matrices Q_1, Q_2, S_1 and S_2 have dimensions $n_2(n_1 + n_2)$, $n_2(n_2 + n_3 + n_4)$, $n_4(n_2 + n_3 +$

n_4) and $n_4(n_4+n_5)$, respectively, and are related to B and D in (4.2) as follows:

$$\begin{aligned}
&Q_1(i,\ j)=B(n_2+i,\ j), && 1\leqslant i\leqslant n_2, \quad 1\leqslant j\leqslant n_1+n_2,\\
&Q_2(i,\ j)=B(i,\ n_1+n_2+j), && 1\leqslant i\leqslant n_2, \quad 1\leqslant j\leqslant n_2+n_3+n_4,\\
&S_1(i,\ j)=D(n_4+i,\ j), && 1\leqslant i\leqslant n_4, \quad 1\leqslant j\leqslant n_2+n_3+n_4,\\
&S_2(i,\ j)=D(i,\ n_2+n_3+n_4+j), && 1\leqslant i\leqslant n_4, \quad 1\leqslant j\leqslant n_4+n_5.
\end{aligned}$$

The vectors t_1, t_2 and t_3 are related to r_1, r_2, r_3, r_4 and r_5 in the following way:

$$\begin{aligned}
&t_1(i)=r_1(i), && 1\leqslant i\leqslant n_1,\\
&t_1(n_1+i)=r_2(i), && 1\leqslant i\leqslant n_2,\\
&t_2(i)=r_2(n_2+i), && 1\leqslant i\leqslant n_2,\\
&t_2(n_2+i)=r_3(i), && 1\leqslant i\leqslant n_3,\\
&t_2(n_2+n_3+i)=r_4(i), && 1\leqslant i\leqslant n_4,\\
&t_3(i)=r_4(n_4+i), && 1\leqslant i\leqslant n_4,\\
&t_3(n_4+i)=r_5(i), && 1\leqslant i\leqslant n_5.
\end{aligned}$$

The system (4.4) may be written in component form as

$$Pa^{\mathrm{I}}+Q_2^*a^{\mathrm{II}}=t_1, \tag{4.5}$$

$$Q_1^*a^{\mathrm{I}}+Ra^{\mathrm{II}}+S_2^*a^{\mathrm{III}}=t_2, \tag{4.6}$$

$$S_1^*a^{\mathrm{II}}+Ta^{\mathrm{III}}=t_3, \tag{4.7}$$

where

$$Q_2^*=\left(\frac{0}{Q_2}\right), \qquad Q_1^*=\left(\frac{Q_1}{0}\right),$$

$$S_2^*=\left(\frac{0}{S_2}\right), \qquad S_1^*=\left(\frac{S_1}{0}\right).$$

The vectors a^{I} and a^{III} may be expressed in terms of a^{II} from (4.5) and (4.7):

$$a^{\mathrm{I}}=P^{-1}t_1-P^{-1}Q_2^*a^{\mathrm{II}}, \tag{4.8}$$

$$a^{\mathrm{III}}=T^{-1}t_3-T^{-1}S_1^*a^{\mathrm{II}}. \tag{4.9}$$

Substitution of (4.8) and (4.9) in (4.6) yields

$$\left(R-Q_1^*P^{-1}Q_2^*-S_2^*T^{-1}S_1^*\right)a^{\mathrm{II}}=t_2-Q_1^*P^{-1}t_1-S_2^*T^{-1}t_3. \tag{4.10}$$

The solution of the above system gives a^{II}. Subsequent substitution into (4.8) and (4.9) yields a^{I} and a^{III}, respectively.

The global system may be solved efficiently using the following algorithm.

Algorithm

(1) Calculate $P^{-1}Q_2^*$ and $P^{-1}t_1$ by solving a system of the form

$$P[J_1\mid w_1]=H_1,$$

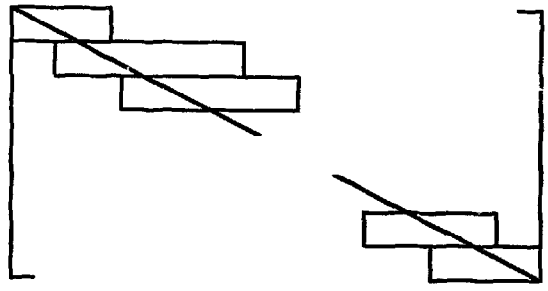

Fig. 3. Almost block diagonal form.

where H_1 is a $(n_1+n_2)(n_2+n_3+n_4+1)$ matrix containing Q_2^* and t_1.

Similarly calculate $T^{-1}S_1^*$ and $T^{-1}t_3$ by solving a system of the form

$$T[J_2 \mid w_2] = H_2,$$

where H_2 is a $(n_4+n_5)(n_2+n_3+n_4+1)$ matrix containing S_1^* and t_3.

(2) Evaluate

$$\tilde{R} = R - Q_1^* J_1 - S_2^* J_2, \qquad \tilde{t}_2 = t_2 - Q_1^* w_1 - S_2^* w_2$$

by exploiting the structures of Q_1^* and S_2^*.

(3) Solve $\tilde{R}a^{\mathrm{II}} = \tilde{t}_2$.

(4) Evaluate $a^{\mathrm{I}} = w_1 - J_1 a^{\mathrm{II}}$, $a^{\mathrm{III}} = w_2 - J_2 a^{\mathrm{II}}$.

4.3. Algorithms for almost block diagonal systems

In this section we examine the application of a production standard code for the solution of almost block diagonal systems [1] to the global system (4.1). This code uses a modified column elimination procedure with alternate row and column pivoting based on an algorithm originally described in [12] and [3] and is intended to solve systems of the form shown in Fig. 3, that is, systems which consist of rectangular blocks along the diagonal and for which no three successive blocks have columns in common. These systems arise naturally in the solution of ordinary differential equation boundary value problems with separated boundary conditions when using finite differences and spline collocation methods.

The global spectral collocation matrix G in (4.1) may be written in almost block diagonal form from its decomposed form (4.4) in the obvious way:

$$G = \left[\begin{array}{c|c|c}
P_1 & 0 & 0 \\ \hline
P_2 & Q_2 & \\ \hline
Q_1 & R_1 & 0 \\ \hline
0 & R_2 & \\ \hline
 & R_3 & S_2 \\ \hline
0 & S_1 & T_1 \\ \hline
 & 0 & T_2
\end{array}\right]. \tag{4.11}$$

G has five nonzero blocks, namely

$$[P_1], \quad \left[\begin{array}{c|c} P_2 & Q_2 \\ \hline Q_1 & R_1 \end{array}\right], \quad [R_2], \quad \left[\begin{array}{c|c} R_3 & S_2 \\ \hline S_1 & T_1 \end{array}\right] \text{ and } [T_2].$$

Unfortunately, the form (4.11) is not that required for the almost block diagonal code due to too much overlap between blocks 2 and 3 and 3 and 4. More specifically the sum of the overlapping columns between the second and third and third and fourth blocks exceeds the number of blocks in the third block (see [8]).

However the transpose of G, from (4.4) yields

$$G^{\mathrm{T}} = \left[\begin{array}{c|c|c|c|c|c} P^{\mathrm{T}} & Q_1^{\mathrm{T}} & & 0 & 0 & \\ \hline 0 & Q_2^{\mathrm{T}} & & R^{\mathrm{T}} & S_1^{\mathrm{T}} & 0 \\ \hline 0 & 0 & & & S_2^{\mathrm{T}} & T^{\mathrm{T}} \end{array}\right], \tag{4.12}$$

which is of the required almost block diagonal form, with three blocks, namely:

$$[P^{\mathrm{T}} | Q_1^{\mathrm{T}}], \quad [Q_2^{\mathrm{T}} | R^{\mathrm{T}} | S_1^{\mathrm{T}}] \text{ and } [S_2^{\mathrm{T}} | T^{\mathrm{T}}].$$

One may, therefore, decompose the transpose of the global matrix, G^{T} using the existing NAG routine F01LHF [8] and subsequently solve for the transpose of the decomposed form of G^{T}, say $\tilde{G}^{\mathrm{T}}$, the system

$$(\tilde{G}^{\mathrm{T}})^{\mathrm{T}} x = r \tag{4.13}$$

with the NAG code F04LHF [8].

One is required to provide the routine F01LHF with a vector form of the matrix G^{T}, read in block by block, column by column.

By defining

$$m_1 = n_1 + n_2, \qquad m_2 = n_2,$$
$$m_3 = n_2 + n_3 + n_4, \qquad m_4 = n_4, \qquad m_5 = n_4 + n_5,$$

from (4.12) the vector form a of G^{T} required by F01LHF is defined as:

$$\begin{aligned}
&a\big(m_1^2(i-1)+j\big) = P(i,\ j), && 1 \leqslant i \leqslant m_1, \quad 1 \leqslant j \leqslant m_1,\\
&a\big(m_1^2 + m_1(i-1)+j\big) = Q_1(i,\ j), && 1 \leqslant i \leqslant m_2, \quad 1 \leqslant j \leqslant m_1,\\
&a\big(m_1^2 + m_1 m_2 + m_3(i-1)+j\big) = Q_2(i,\ j), && 1 \leqslant i \leqslant m_2, \quad 1 \leqslant j \leqslant m_3,\\
&a\big(m_1^2 + m_1 m_2 + m_2 m_3 + m_3(i-1)+j\big) = R(i,\ j), && 1 \leqslant i \leqslant m_3, \quad 1 \leqslant j \leqslant m_3,\\
&a\big(m_1^2 + m_1 m_2 + m_2 m_3 + m_3^2 + m_3(i-1)+j\big) = S_1(i,\ j), && 1 \leqslant i \leqslant m_4, \quad 1 \leqslant j \leqslant m_3,\\
&a\big(m_1^2 + m_1 m_2 + m_2 m_3 + m_3^2 + m_3 m_4 + m_5(i-1)+j\big) = S_2(i,\ j),\\
& && 1 \leqslant i \leqslant m_4, \quad 1 \leqslant j \leqslant m_5,\\
&a\big(m_1^2 + m_1 m_2 + m_2 m_3 + m_3^2 + m_3 m_4 + m_4 m_5 + m_5(i-1)+j\big) = T(i,\ j),\\
& && 1 \leqslant i \leqslant m_5, \quad 1 \leqslant j \leqslant m_5.
\end{aligned}$$

4.4. Computational cost

The solution of the full system (4.1) requires $kN^3 + N^2$ operations where $N = n_1 + n_3 + n_5 + 2(n_2 + n_4)$.

In the solution of (4.3) by a capacitance matrix technique most of the work lies in solving four systems of equations of orders n_1, n_3, n_5 and $2(n_2 + n_4)$ with multiple right-hand sides (see [11]) requiring $k(n_1)^3 + n_2 n_1^2$, $k(n_3)^3 + (n_2 + n_4)n_3^2$, $k(n_5)^3 + n_4 n_5^2$ and $8k(n_2 + n_4)^3 + 4(n_2 + n_4)^2$ operations, respectively. Since n_1, n_3 and n_5 are in practice much larger than $n_2 + n_4$, this technique produces savings of $O(n_1^2(n_3 + n_5) + n_3^2(n_1 + n_5) + n_5^2(n_1 + n_3))$ over the solution of the full system.

In the algorithm for the solution of system (4.4) most of the work is expended in steps (1) and (3). Step (1) requires $k(n_1 + n_2)^3 + (n_2 + n_3 + n_4 + 1)(n_1 + n_2)^2 + k(n_4 + n_5)^3 + (n_2 + n_3 + n_4 + 1)(n_4 + n_5)^2$ operations and step (3) requires $k(n_2 + n_3 + n_4)^3 + (n_2 + n_3 + n_4)^2$ operations. This totals approximately $O(n_3(n_1^2 + n_5^2))$ more operations than the capacitance matrix technique.

Most of the work in the solution of the system (4.13) is expended in the decomposition of the almost block diagonal matrix (4.12) with the routine F01LHF. This requires (see [5]):

$$\sum_{j=1}^{n_1} (n_1 + 2n_2 - j)(n_1 + n_2 - j) + \sum_{j=1}^{n_2} (2n_2 + n_3 + n_4 - j)(2n_2 - j)$$
$$+ \sum_{j=1}^{n_2+n_3} (n_2 + n_3 + 2n_4 - j)(n_2 + n_3 + n_4 - j) + \sum_{j=1}^{n_4} (2n_4 + n_5 - j)(2n_4 - j)$$
$$+ \sum_{j=1}^{n_4+n_5} (n_4 + n_5 - j)^2$$

flops. Taking into consideration that n_2 and n_4 are much smaller than n_1, n_3 and n_5 this gives approximately

$$k\left(n_1^3 + n_3^3 + n_5^3\right) + \tfrac{3}{2}n_1^2 n_2 + n_3^2\left(\tfrac{3}{2}n_4 + n_2\right) + n_5^2 n_4,$$

which is almost the same as the number of operations required by the capacitance matrix technique.

5. Numerical results

Apart from the obvious savings in storage provided by the three techniques described in Section 4 over solving the full system they also provide considerable savings in the computational cost of solving the problem in question. We examine five cases involving different numbers of degrees of freedom. In each case we choose the highest degrees of the Chebyshev polynomials in each direction and each element to be equal to N. Table 1 contains the CPU-times in seconds for each of the cases $N = 9$, 11, 13, 15 and 17 for the solution of the full system without taking into account the sparsity of the matrix, the coefficient splitting technique, the capacitance matrix technique and the application of the NAG routine F01LHF in decomposing the transpose of the original system. The capacitance matrix technique and F01LHF always perform better than the

Table 1
CPU (IBM 3081-D32) times (in seconds) required to solve the 2:1 contraction problem for different numbers of degrees of freedom

N	Total degrees of freedom	Full system	Coefficient splitting	F01LHF	Capacitance technique
9	176	3.92	4.16	3.12	2.38
11	280	9.36	9.44	5.57	4.70
13	408	22.70	21.73	10.51	10.12
15	560	51.90	47.12	19.91	20.27
17	736	109.33	92.01	36.34	38.76

Table 2
Condition numbers for the three types of system matrices for different numbers of degrees of freedom

N	Total degrees of freedom	Full system (G)	Coefficient splitting (R)	Capacitance technique ($\tilde{F}$)
9	176	0.1175(8)	0.1604(6)	0.4146(11)
11	280	0.6903(8)	0.6517(6)	0.3686(12)
13	408	0.2981(9)	0.2051(7)	0.2036(14)

coefficient splitting technique and their efficiency grows with the number of degrees of freedom used. As the number of degrees of freedom grows, F01LHF performs slightly better than the capacitance matrix technique. These savings become greater when solving nonlinear problems,

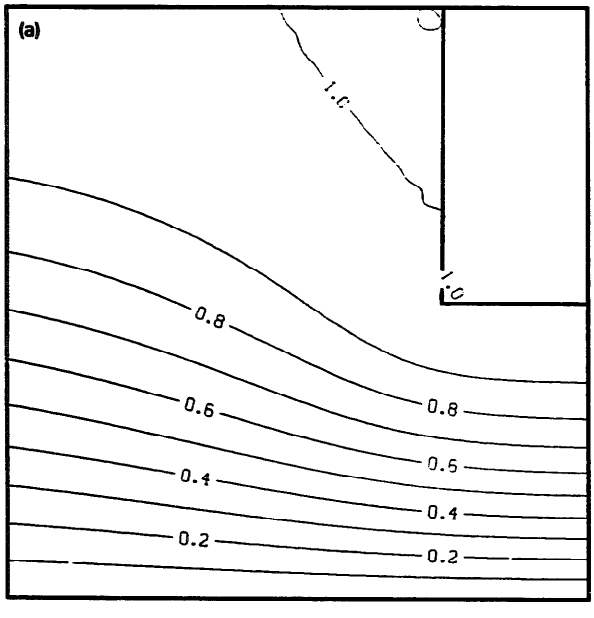

Fig. 4(a). Streamfunction contours for $\alpha = \frac{1}{2}$ on $-0.75 \leqslant x \leqslant 0.25$.

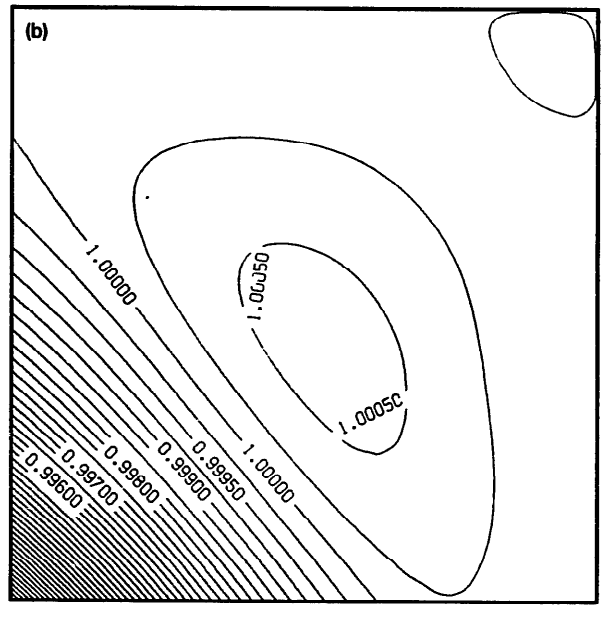

Fig. 4(b). Streamfunction contours in salient corner for $\alpha = \frac{1}{2}$ on $-0.25 \leqslant x \leqslant 0, 0.75 \leqslant y \leqslant 1$.

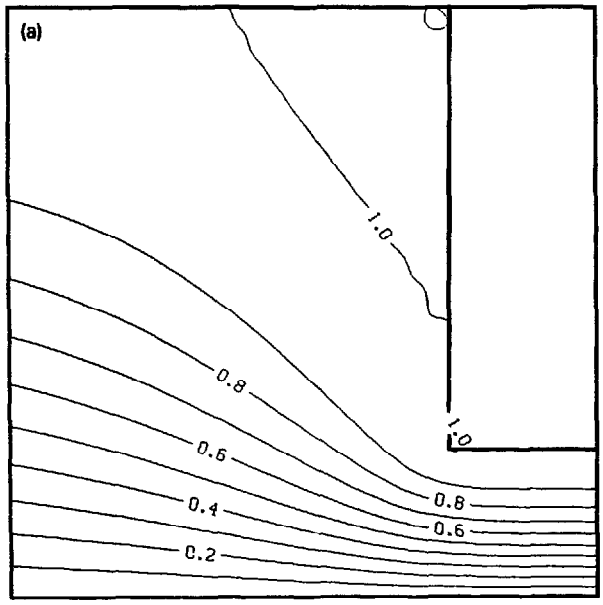

Fig. 5(a). Streamfunction contours for $\alpha = \frac{1}{4}$ on $-0.75 \leqslant x \leqslant 0.25$.

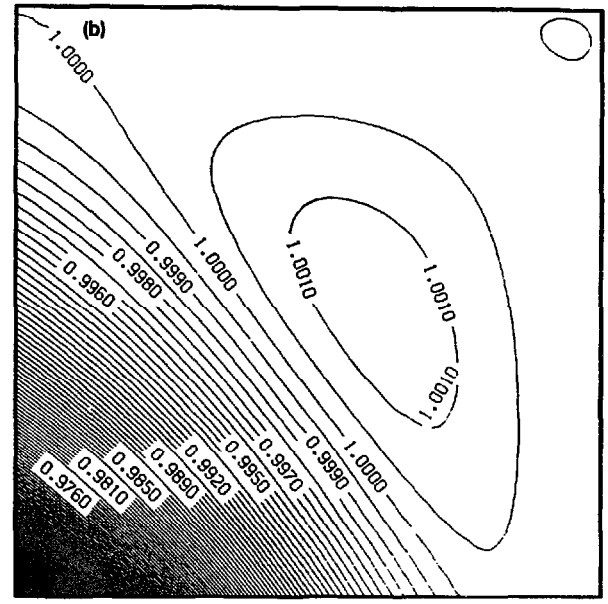

Fig. 5(b). Streamfunction contours in salient corner for $\alpha = \frac{1}{4}$ on $-0.4 \leqslant x \leqslant 0, 0.6 \leqslant y \leqslant 1$.

such as the Navier–Stokes equations where similar systems need to be solved at each step of an iterative process.

The conditioning of the systems in each case was also investigated. In Table 2, we present the condition numbers (infinity norm) of the matrices G (full system and F01LHF), R (coefficient

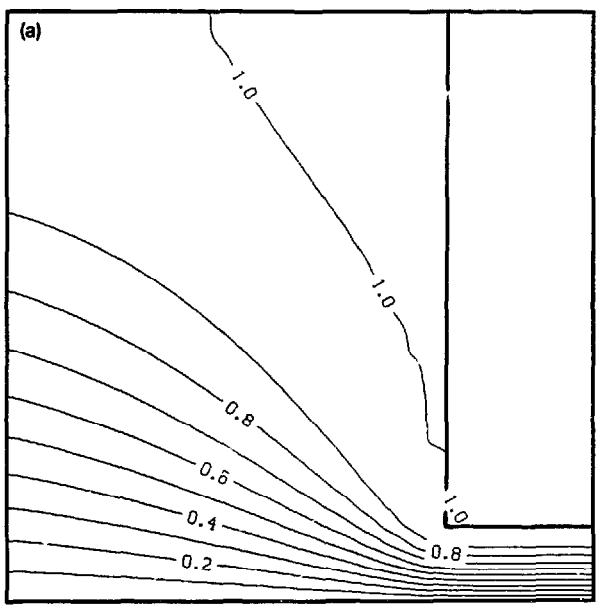

Fig. 6(a). Streamfunction contours for $\alpha = \frac{1}{8}$ on $-0.75 \leqslant x \leqslant 0.25$.

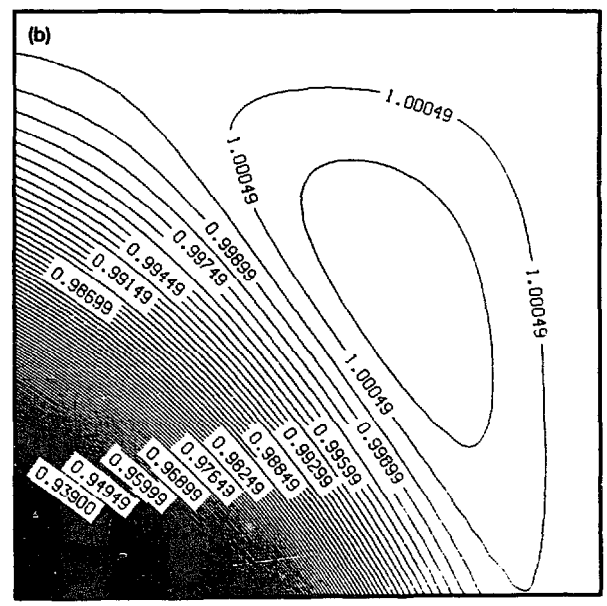

Fig. 6(b). Streamfunction contours in salient corner for $\alpha = \frac{1}{8}$ on $-0.5 \leqslant x \leqslant 0, 0.5 \leqslant y \leqslant 1$.

splitting technique) and $\tilde{F}$ (capacitance matrix) for various numbers of degrees of freedom. It is clear that the systems resulting from the capacitance matrix technique are very poorly conditioned in comparison to the systems involved in the coefficient splitting technique or the global matrix. In some instances, the capacitance matrix technique produced numerically singular matrices for large numbers of degrees of freedom. The poor conditioning of Chebyshev systems [4] is evidently magnified when global pivoting is not performed. The NAG routine F01LHF on the other hand performs alternate global row and column pivoting and after extensive experimentation no singular cases were encountered.

Further, contour plots for the solution for different values of the contraction ratio α, are given. In Fig. 4, we present contours of the stream function for $\alpha = \frac{1}{2}$. In this case $h_1 = 1$, $h_2 = 0.5$, $M^{\mathrm{I}} = M^{\mathrm{II}} = 20$, $M^{\mathrm{III}} = 8$, $N^{\mathrm{I}} = 10$, $N^{\mathrm{II}} = N^{\mathrm{III}} = 14$ giving a total of 495 degrees at freedom. In Fig. 5 we present contours of the stream function for $\alpha = \frac{1}{4}$ obtained with $h_1 = 1.5$, $h_2 = 1.0$, $M^{\mathrm{I}} = M^{\mathrm{II}} = 26$, $M^{\mathrm{III}} = 14$, $N^{\mathrm{I}} = 20$, $N^{\mathrm{II}} = N^{\mathrm{III}} = 8$, totalling 715 degrees of freedom. Finally in Fig. 6 we present contours of the stream function for $\alpha = \frac{1}{8}$ obtained with $h_1 = 1.5$, $h_2 = 1.0$, $M^{\mathrm{I}} = M^{\mathrm{II}} = 26$, $M^{\mathrm{III}} = 14$, $N^{\mathrm{I}} = 20$, $N^{\mathrm{II}} = N^{\mathrm{III}} = 8$, totalling 715 degrees of freedom.

6. Conclusions

Spectral domain decomposition methods for solving the biharmonic equation in rectangularly decomposable domains results in a linear system of equations with a coefficient matrix that is block tridiagonal. Direct methods which exploit this matrix structure are described. The following methods are compared with respect to computational cost, storage and conditioning: the capacitance matrix method, the coefficient splitting method and a NAG routine for almost block diagonal systems. The coefficient matrix of the transposed system is in this form and so is amenable to solution by this technique.

For fairly moderate numbers of degrees of freedom the capacitance matrix method and the NAG routine are comparable in terms of computational cost and storage. Both of these methods perform better than the coefficient splitting method. For problems in which a large number of degrees of freedom are required to compute the solution accurately, the capacitance matrix method suffers from poor conditioning and, in some cases, produces numerically singular submatrices. No such problems are found for the NAG routine.

All the techniques described in this paper are applicable to the solution of nonlinear partial differential equations, such as the Navier–Stokes equations, using spectral domain decomposition techniques. The coefficient matrix of the linearized equations has the same structure as those considered in this paper. The savings in CPU-time are even greater for nonlinear problems since such systems of equations need to be solved at each stage of a linearization process.

Acknowledgement

The authors are grateful to R.W. Brankin for his useful comments.

References

[1] R.W. Brankin and I. Gladwell, Codes for almost block diagonal systems, *Comput. Math. Appl.* **19** (1990) 1–6.
[2] B.L. Buzbee, F.W. Dorr, J.A. George and G.H. Golub, The direct solution of the discrete Poisson equation on irregular regions, *SIAM J. Numer. Anal.* **8** (1971) 722–736.
[3] J.C. Diaz, G. Fairweather and P. Keast, FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column eliminations, *ACM Trans. Math. Software* **9** (1983) 358–375.
[4] D. Gottlieb and S.A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications* (SIAM-CBMS, Philadelphia, PA, 1977).
[5] R.I. Hay, The impact of vector processors on boundary value codes, Ph.D. Thesis, Univ. Manchester, 1986.
[6] A. Karageorghis and T.N. Phillips, Spectral collocation methods for Stokes flow in contraction geometries and unbounded domains, *J. Comput. Phys.* **80** (1989) 314–330.
[7] M.G. Macaraeg and C.L. Streett, Improvements in spectral collocation discretization through a multiple domain technique, *Appl. Numer. Math.* **2** (2) (1986) 95–108.
[8] Numerical Algorithms Group Library Mark 13, NAG (UK) Ltd, Jordan Hill Road, Oxford, UK.
[9] A.T. Patera, Fast direct Poisson solvers for high-order finite element discretizations in rectangularly-decomposable domains, *J. Comput. Phys.* **65** (1986) 474–480.
[10] T.N. Phillips and A. Karageorghis, Efficient direct methods for Stokes flow in rectangularly decomposable domains, *SIAM J. Sci. Statist. Comput.* **10** (1989) 89–103.
[11] T.N. Phillips and A. Karageorghis, A conforming spectral collocation strategy for Stokes flow through a channel contraction, *Appl. Numer. Math.*, to appear.
[12] J.M. Varah, Alternate row and column elimination for solving certain linear systems, *SIAM J. Numer. Anal.* **13** (1976) 71–75.