Numer. Math. 31, 49–61 (1978)

Numerische Mathematik

Efficient Algorithms for Solving Tensor Product Finite Element Equations

Randolph E. Bank

Department of Mathematics, University of Texas at Austin,
Austin, TX 78712, USA

Summary. There are currently several highly efficient methods for solving linear systems associated with finite difference approximations of Poisson's equation in rectangular regions. These techniques are employed to develop both direct and iterative methods for solving the linear systems arising from the use of C^0 quadratic or C^1 cubic tensor product finite elements.

Subject Classifications. AMS (MOS): 65N20; CR: 5.17.

1. Introduction

In recent years, a number of efficient algorithms, known as *fast direct methods*, have been proposed for solving elliptic partial difference equations [1–3, 7]. The algorithms are usually discussed in terms of solving the linear systems associated with 5-point finite difference equations, and require $O(n^2)$ to $O(n^2 \log n)$ arithmetic operations to solve a problem on an $n \times n$ grid. Most of these methods can be extended to cover the finite element matrices arising from the use of tensor product C^0 linear finite elements, since these matrices can be viewed in terms of 9-point finite difference matrices [12].

In this work, we develop both direct and iterative methods for solving the linear systems arising from the use of tensor product C^0 quadratic and C^1 cubic finite elements. These methods rely heavily on the techniques developed for finite difference equations. We consider the solution of the model problem

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega=(0,1)\times(0,1) \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{1.1}$$

Let $h=\dfrac{1}{n+1}$ characterize a uniform mesh on Ω and let $(x_i, y_i)=(ih, jh)$, $0 \leqq i,j \leqq n+1$. Let

$$\begin{aligned} p_1(x) &= \begin{cases} 1-x & 0 \leqq x \leqq 1 \\ 1+x & -1 \leqq x < 0; \\ 0 & \text{otherwise}; \end{cases} \\ p_2(x) &= \begin{cases} x(1-x)/2 & 0 \leqq x \leqq 1 \\ 0 & \text{otherwise}; \end{cases} \end{aligned} \tag{1.2}$$

and define $\phi_i(x)$, $1 \leqq i \leqq 2n+1$ by

$$\begin{aligned} \phi_{2i}(x) &= p_1\left(\frac{x-x_i}{h}\right), && 1 \leqq i \leqq n; \\ \phi_{2i+1}(x) &= -h^2 p_2\left(\frac{x-x_i}{h}\right), && 0 \leqq i \leqq n. \end{aligned} \tag{1.3}$$

Then a basis for the tensor product C^0 quadratic finite element subspace relevant to the solution of (1.1) is

$$\{\phi_i(x)\}_{i=1}^{2n+1} \times \{\phi_j(y)\}_{j=1}^{2n+1}. \tag{1.4}$$

Let

$$\begin{aligned} p_3(x) &= \begin{cases} (1-x)^2(1+2x) & 0 \leqq x \leqq 1 \\ (1+x)^2(1-2x) & -1 \leqq x < 0 \\ 0 & \text{otherwise,} \end{cases} \\ p_4(x) &= \begin{cases} x(1-x)^2 & 0 \leqq x \leqq 1 \\ x(1+x)^2 & -1 \leqq x < 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{1.5}$$

and define $\psi_i(x)$, $1 \leqq i \leqq 2n+2$ by

$$\psi_1(x) = \begin{cases} h\, p_4\left(\frac{x}{h}\right), & x \geqq 0 \\ 0, & \text{otherwise;} \end{cases} \tag{1.6}$$

$$\psi_{2i}(x) = p_3\left(\frac{x-x_i}{h}\right), \qquad 1 \leqq i \leqq n$$

$$\psi_{2i+1}(x) = h\, p_4\left(\frac{x-x_i}{h}\right), \qquad 1 \leqq i \leqq n;$$

$$\psi_{2n+2}(x) = \begin{cases} h\, p_4\left(\frac{x-1}{h}\right), & x \geqq 1 \\ 0, & \text{otherwise.} \end{cases}$$

Then a basis for the tensor product C^1 cubic finite element subspace relevant to (1.1) is

$$\{\psi_i(x)\}_{i=1}^{2n+2} \times \{\psi_j(y)\}_{j=1}^{2n+2}. \tag{1.7}$$

The application of the finite element method to (1.1) leads to a linear system of equations

$$Mx = b \tag{1.8}$$

where M is large and sparse (of order $(2n+1)^2$ for the C^0 quadratic subspace and $(2n+2)^2$ for the C^1 cubic subspace). In either case the matrix M is

symmetric, positive definite, and is characterized by the tensor product structure

$$M = T_m \otimes T_s + T_s \otimes T_m \tag{1.9}$$

where T_m and T_s are 1-dimensional mass and stiffness matrices respectively and $\otimes$ denotes matrix tensor product.

In Section 2, we consider the direct solution of (1.8) using methods based on the fast Fourier transform. These algorithms require $O(n^2 \log n)$ operations to solve the linear system. In Section 3, we discuss several block iterative methods employing fast algorithms developed for 9-point finite difference equations. The spectral radii of these iterations is shown to be independent of n; thus they require $O(n^2 \log 1/\varepsilon)$ to $O(n^2 \log n \log 1/\varepsilon)$ operations, depending on which fast method used, to reduce the initial error by a factor of ε.

In Section 4, we describe enhancements to the iterative methods which allow the use of software appropriate to 5-point finite difference equations. We make some concluding remarks in Section 5.

2. Direct Methods Based on the Fast Fourier Transform

In this section we develop fast algorithms for solving (1.8) for both C^0 quadratic and C^1 cubic finite element subspaces described in Section 1. Our methods employ the fast sine and fast cosine transforms, both computable using standard FFT routines [6, 13]. We define the sine transform of an m-vector v as the matrix-vector product $\hat{v} = S v$, where

$$S_{ij} = \sqrt{\frac{2}{m+1}} \sin\left(\frac{i\pi j}{m+1}\right) \qquad 1 \leqq i,j \leqq m. \tag{2.1}$$

The cosine transform of an $m+2$ vector w is defined as the matrix-vector product $\hat{w} = C w$, where

$$C_{i+1\,j+1} = \sqrt{\frac{2}{m+1}}\, e_j \cos\left(\frac{i\pi j}{m+1}\right) \qquad 0 \leqq i,j \leqq m+1, \tag{2.2}$$

where $e_j = 1/2$ for $j = 0$, $m+1$ and $e_j = 1$ otherwise. It is straightforward to verify $S^{-1} = S$, and $C^{-1} = C$, so that the inverse (synthesis) transforms are the same as the analysis transforms.

We first consider the C^0 quadratic finite element space of Section 1. The mass and stiffness matrices are $(2n+1) \times (2n+1)$, penta-diagonal, and of the form

$$T = \begin{bmatrix} d & e & 0 & & & & & & \\ b & a & b & c & & & & & \\ 0 & e & d & e & 0 & & & & \\ & c & b & a & b & c & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & 0 & e & d & e & 0 & \\ & & & & c & b & a & b & \\ & & & & & 0 & e & d & \end{bmatrix}. \tag{2.3}$$

For the mass matrix T_m, $a=2h/3$, $b=e=-h^3/24$, $c=h/6$, and $d=h^5/120$. For the stiffness matrix T_s, $a=2/h$, $b=e=0$, $c=-1/h$, and $d=h^3/12$.

We apply the sine transform of order $2n+1$ to the matrix T; that is, we form the product $\bar{T}=STS^{-1}$. From (2.1), it is easily verified that

$$\begin{aligned}\bar{T}_{ij}&=\frac{1}{n+1}\sum_{l=1}^{2n+1}\sum_{k=1}^{2n+1}\sin\left(\frac{i\pi k}{2n+2}\right)T_{kl}\sin\left(\frac{l\pi j}{2n+2}\right)\\&=\frac{1}{2}\left\{a+2b\cos\left(\frac{j\pi}{2n+2}\right)+2c\cos\left(\frac{j\pi}{n+1}\right)\right.\\&\quad\left.-d-2e\cos\left(\frac{j\pi}{2n+2}\right)\right\}(\delta_{ij}-\delta_{2n+2-ij})\\&\quad+\left\{d+2e\cos\left(\frac{j\pi}{2n+2}\right)\right\}\delta_{ij},\end{aligned}\tag{2.4}$$

where δ_{ij} is the Kronecker delta.

From (2.4) it follows that the matrix $\bar{T}$ can be recordered to become block diagonal with n 2×2 blocks and a single 1×1 block. It is convenient to apply the orthogonal transformation $Q=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}$ to each of the 2×2 blocks. This yields the final transformed matrix $\hat{T}$ given by

$$\hat{T}=\begin{bmatrix}\boxed{\begin{matrix}a+2c\cos\left(\frac{\pi}{n+1}\right) & 2b\cos\left(\frac{\pi}{2n+2}\right)\\ 2e\cos\left(\frac{\pi}{2n+2}\right) & d\end{matrix}} & & & \bigcirc\\ & \ddots & & \\ & & \boxed{\begin{matrix}a+2c\cos\left(\frac{n\pi}{n+1}\right) & 2b\cos\left(\frac{n\pi}{2n+2}\right)\\ 2e\cos\left(\frac{n\pi}{2n+2}\right) & d\end{matrix}} & \\ \bigcirc & & & \boxed{d}\end{bmatrix}\tag{2.5}$$

In the C^1 cubic case, the mass and stiffness matrices are of order $2n+2$, block tridiagonal, with 2×2 blocks, and have the form

$$T=\left[\begin{array}{c|cc|cc|cc|cc|cc|c}
d/2 & e & f & & & & & & & & & \\
b & a & 0 & c & -b & & & & & & & \\
f & 0 & d & e & f & & & & & & & \\
 & c & b & a & 0 & c & -b & & & & & \\
 & -e & f & 0 & d & e & f & & & & & \\
 & & \ddots & & \ddots & & \ddots & & & & & \\
 & & & & & c & b & a & 0 & c & -b & \\
 & & & & & -e & f & 0 & d & e & f & \\
 & & & & & & & c & b & a & 0 & -b \\
 & & & & & & & -e & f & 0 & d & f \\
 & & & & & & & & & -e & f & d/2
\end{array}\right] \tag{2.6}$$

For the mass matrix T_m, $a=26h/35$, $b=e=13h^2/420$, $c=9h/70$, $d=2h^3/105$, and $f=-h^3/140$. For the stiffness matrix T_s, $a=12/5h$, $b=e=-1/10$, $c=-6/5h$, $d=4h/15$ and $f=-h/30$. For our present purpose, it is convenient to multiply the first and last rows of T by 2, and then recorder to form the matrix T' given by

$$T'=\left[\begin{array}{cccccc|cccccc}
a & c & & & & & b & 0 & -b & & & \\
c & a & c & & & & & b & 0 & -b & & \\
 & \ddots & \ddots & \ddots & & & & & \ddots & \ddots & \ddots & \\
 & & c & a & c & & & & & b & 0 & -b \\
 & & & c & a & & & & & & b\;\;0 & -b \\
\hline
2e & & & & & & d & 2f & & & & \\
0 & e & & & & & f & d & f & & & \\
-e & 0 & \ddots & & & & & \ddots & \ddots & \ddots & & \\
 & -e & \ddots & e & & & & & & & & \\
 & & \ddots & 0 & e & & & & f & d & f & \\
 & & & -e & 0 & & & & & & & \\
 & & & & -2e & & & & & & 2f & d
\end{array}\right] \tag{2.7}$$

Consistent with the partitioning of T', we define the matrix

$$Q=\begin{bmatrix} S & 0 \\ 0 & C \end{bmatrix} \tag{2.8}$$

where S is $n\times n$ and C is $(n+2)\times(n+2)$, as defined in (2.1)–(2.2). Clearly $Q^{-1}=Q$. We form the product $QT'Q^{-1}$ and recorder, yielding the transformed matrix $\hat{T}$

given by

$$\hat{T}=\begin{bmatrix} d+2f & & & & \\ & \begin{matrix} a+2c\cos\left(\frac{\pi}{n+1}\right) & 2b\sin\left(\frac{\pi}{n+1}\right) \\ 2e\sin\left(\frac{\pi}{n+1}\right) & d+2f\cos\left(\frac{\pi}{n+1}\right) \end{matrix} & & \bigcirc & \\ & & \ddots & & \\ & \bigcirc & & \begin{matrix} a+2c\cos\left(\frac{n\pi}{n+1}\right) & 2b\sin\left(\frac{n\pi}{n+1}\right) \\ 2e\sin\left(\frac{n\pi}{n+1}\right) & d+2f\cos\left(\frac{n\pi}{n+1}\right) \end{matrix} & \\ & & & & d-2f \end{bmatrix} \tag{2.9}$$

With the transformed matrices (2.5) and (2.9) it is straightforward to describe FFT methods for solving (1.8). Let M be defined as in (1.9). Then to solve (1.8), take appropriate Fourier transforms in both directions, yielding the linear system

$$\hat{M}\hat{x}=\hat{b}$$
$$\hat{M}=\hat{T}_m\otimes\hat{T}_s+\hat{T}_s\otimes\hat{T}_m. \tag{2.10}$$

For both finite element subspaces under consideration, $\hat{M}$ can be recorded to become block diagonal; for the C^0 quadratics, there are n^2 4×4 blocks, $2n$ 2×2 blocks and a single 1×1 block. For the C^1 cubics, there are n^2 4×4 blocks, $4n$ 2×2 blocks, and 4 1×1 blocks.

The FFT algorithm can be summarized as

(1) Take appropriate FFT's in both directions;
(2) Solve the block diagonal system (2.10);
(3) Take inverse FFT's in both directions.

Steps (1) and (3) both require $O(n)$ calls to standard FFT routines. Since the FFT requires $O(n\log n)$ operations, these steps require $O(n^2\log n)$ operations. Solving n^2 4×4 systems requires $O(n^2)$ operations; the remaining diagonal blocks in (2.10) can be handled in $O(n)$ operations. Thus step (2) requires $O(n^2)$ operations, for an overall operation count of $O(n^2\log n)$.

The matrix decomposition algorithm [3, 7] can be extended to cover (1.7). In this algorithm, FFT's are applied in only one space dimension. We must therefore solve a linear system involving the matrix

$$\overline{M}=\hat{T}_s\otimes T_m+\hat{T}_m\otimes T_s.$$

For both subspaces, the matrix $\overline{M}$ can be permuted to be block diagonal, with $O(n)$ diagonal blocks of fixed bandwidth, i.e., independent of n. Thus the linear

system involving $\overline{M}$ can be solved in $O(n^2)$ operations. The matrix decomposition is asymptotically faster than the first FFT algorithm, since only half as many FFT's are required.

3. Block Iterative Methods

Consider the linear system $MU=B$, recorded and partitioned such that

$$MU=\begin{bmatrix} D_1 & E_{12} & E_{13} & E_{14} \\ E_{12}^T & D_2 & E_{23} & E_{24} \\ E_{13}^T & E_{23}^T & D_3 & E_{34} \\ E_{14}^T & E_{24}^T & E_{34}^T & D_4 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}. \tag{3.1}$$

In the case of C^0 quadratics, the unknowns U_1 can be associated with approximations of the function u of (1.1), U_2 corresponds to u_{xx}, U_3 to u_{yy} and U_4 to u_{xxyy}. For the C^1 cubic elements, U_1 corresponds to u, U_2 to u_x, U_3 to u_y and U_4 to u_{xy}. Viewing (3.1) as a system of *finite difference* equations, in both the C^0 quadratic and C^1 cubic cases, D_1 arises from a 9-point approximation of $-c\Delta$, c a constant. For C^0 quadratic elements, D_2 and D_3 result from 3-point approximations of a scalar constant, and D_4 from a one-point approximation of a constant. For C^1 cubic elements, D_i, $i=2,3,4$ result from 9-point approximations of scalar constants. This is seen by formally taking the limit of the difference equations as $h \to 0$ and applying Taylor's theorem.

All linear systems of the form $D_i x = y$ can be efficiently solved, using fast direct methods for finite difference equations if D_i arises from a 9-point discretization, and the well-known algorithm for tridiagonal matrices [9] if D_i arises from a 3-point approximation, thus we are led to the study of block iterative methods based on this partitioning of the matrix M. Let

$$M=-E^T+D-E \tag{3.2}$$

where $D=\mathrm{Diag}[D_i]$ is block diagonal and E is block upper triangular. Let x_0 be a given initial vector, and consider the solution of (1.8) using the following block iterative methods [14].

Block Jacobi: $D(x_{k+1}-x_k)=b-Mx_k$, $k=0,1,\ldots$

Block Gauss Seidel: $(D-E^T)(x_{k+1}-x_k)=b-Mx_k$, $k=0,1,\ldots$

Block Successive Over-Relaxation (SOR):

$$(D-\omega E^T)(x_{k+1}-x_k)=\omega(b-Mx_k), \qquad k=0,1,\ldots. \tag{3.3}$$

Theorem 3.1. Let M denote the tensor product matrix in (3.1) for either C^0 quadratic or C^1 cubic tensor product finite element subspaces defined in (1.2)–(1.7). Then the block iterative methods (3.3) are all convergent (the block SOR for $0<\omega<2$). In particular, the spectral radii of the iteration matrices are essentially independent of n, and are given in Table 3.1.

Table 3.1. Spectral radii for the block iterative methods (3.3)

	C^0 Quadratics	C^1 Cubics
Block Jacobi	$\sqrt{5/6+O(h^2)}$	$\sqrt{1/6+O(h^2)}$
Block Gauss Seidel	$\frac{5}{6}+O(h^2)$	$\frac{1}{6}+O(h^2)$
Block SOR (at ω_{opt})	$0.4202\ldots+O(h^2)$	$0.0455\ldots+O(h^2)$
ω_{opt}	$1.4202\ldots+O(h^2)$	$1.0455\ldots+O(h^2)$

Proof. The convergence of the methods is readily established using standard results [14, Section 3.4], since the matrix M is real, symmetric, and positive definite [11, 12]. To compute the spectral radii, we apply the appropriate Fourier transforms described in Section 2 to the iterations (3.3). After recording, we find that in the transformed coordinates, the problem is reduced to the study of n^2 4×4 iterations and $O(n)$ 2×2 iterations. The results of Table 3.1 follow from the evaluation of the relevant 4×4 determinants.

As an illustration, consider the block Jacobi iteration for C^0 quadratics. In this case we seek λ satisfying

$$\text{Det}\begin{bmatrix} a\lambda & -b & -b & 0 \\ -b & r\lambda & 0 & -g \\ -b & 0 & r\lambda & -g \\ 0 & -g & -g & f\lambda \end{bmatrix} = 0 \tag{3.4}$$

where $a=8s^2(3-2s^2)/3$, $b=h^2 s\,c/3$, $r=h^4(15-4s^2)/180$, $g=h^6 c/144$, $f=h^8/720$, $c=\cos\left(\frac{\pi}{2n+2}\right)$, and $s=\sin\left(\frac{\pi}{2n+2}\right)$. Standard algebraic manipulations reduce (3.4) to

$$\text{Det}\begin{bmatrix} \lambda & 0 & p & 0 \\ 1 & \lambda & 0 & q \\ 0 & -\lambda & \lambda & 0 \\ 0 & 0 & 1 & \lambda \end{bmatrix} = 0 \tag{3.5}$$

where $p=2b^2/(ar)$ and $q=2g^2/(fr)$. From (3.5) we have

$$\lambda^4-\lambda^2(p+q)=0. \tag{3.6}$$

The solutions of (3.4) are thus 0 and $\pm\sqrt{p+q}=\pm\sqrt{\frac{5}{6}}\left(\frac{c}{\sqrt{1-\frac{2}{3}s^2}}\right)=\pm\sqrt{\frac{5}{6}}+O(h^2)$.

The computations in the remaining cases follow a similar pattern. In the case of C^0 quadratics, the 4×4 matrix is a consistently ordered 2-cyclic Stieltjes matrix [14], so much of the established theory for iterative methods is applicable. This is not true in the C^1 cubic case, except in the limit as $h\to 0$, when the relevant 4×4 matrix becomes reducible, yielding 3×3 and 1×1 irreducible blocks to which the standard

theory applies. For the C^1 cubics, perturbation theory for polynomial zeroes [15] appears to be more useful in establishing the results of Theorem 3.1.

The importance of Theorem 3.1 lies in the fact that the spectral radii for these block iterations are essentially independent of n; thus the number of iterations required to reduce the initial error by a factor of ε is also independent of n, being proportional to $-\log\varepsilon$. This is in contrast to the corresponding "point" iterations, where the number of iterations is proportional to $-n\log\varepsilon$. Since solving linear systems of the form $D_i x = y$ requires $O(n^2)$ to $O(n^2 \log n)$ operations, depending on which fast direct methods are employed, the overall costs of these iterative schemes is $O(n^2 \log 1/\varepsilon)$ to $O(n^2 \log n \log 1/\varepsilon)$ computations.

We make one final remark on implementation. Often in practice one scales the basis functions by appropriate powers of h, such that the matrix elements are all of the same order of magnitude. In this event, the unknowns do not correspond identically to function and derivative values but rather to scalar multiples of them. Scalings of this type do not alter the results of Theorem 3.1.

4. Inner Iterations

To motivate our discussion of inner iterations, we first consider the tensor product matrices arising from finite difference approximations of $-c \cdot \Delta$, c a scalar. Let T be the $n \times n$ tridiagonal matrix with diagonal entries 2 and off-diagonal entries -1, which we denote $T = [-1 \;\; 2 \;\; -1]$, and let T' be the $n \times n$ tridiagonal matrix $T' = \left[\frac{b}{2} \;\; a+b \;\; \frac{b}{2}\right]$, $a \geqq 0$, $b \geqq 0$. Define D_1 by

$$D_1 = T \otimes T' + T' \otimes T. \tag{4.1}$$

The matrix D_1 arises from 9-point finite difference approximations of $-c\Delta$, for example, the matrices D_1 of (3.1). Define G_1 by

$$G_1 = (a+b)\, T \otimes I + I \otimes T) \tag{4.2}$$

where I is the identity matrix of order n. The matrix G_1 results from standard 5-point difference approximations of $-c\Delta$. We wish to solve the linear system $D_1 x = y$ using the iteration

$$G_1(x_{k+1} - x_k) = \omega(y - D_1 x_k), \qquad x_0 \text{ given}, \quad k = 0, 1, \ldots. \tag{4.3}$$

It is convenient to analyze the iteration (4.3) in its transformed coordinates; in this case, sine transforms are used in both directions. The transformed matrices $\hat{D}_1$ and $\hat{G}_1$ are diagonal. The values of λ satisfying

$$\mathrm{Det}(\lambda \hat{G}_1 - \hat{G}_1 + \omega \hat{D}_1) = 0 \tag{4.4a}$$

are given by

$$\lambda_{ij}=1-\omega-\frac{\omega b}{(a+b)}\left(\frac{\cos\left(\frac{i\pi}{n+1}\right)+\cos\left(\frac{j\pi}{n+1}\right)-2\cos\left(\frac{i\pi}{n+1}\right)\cos\left(\frac{j\pi}{n+1}\right)}{2-\cos\left(\frac{i\pi}{n+1}\right)-\cos\left(\frac{j\pi}{n+1}\right)}\right), \tag{4.4b}$$

$1\leqq i,j\leqq n.$

From (4.4), we find the optimum value of ω is 1, and that the spectral radius is $\frac{b\cos\left(\frac{n}{n+1}\right)}{a+b}=\frac{b}{a+b}+O(h^2)$. For the matrix D_1 associated with C^0 quadratics, $a=b=1/3$, yielding a spectral radius of $\frac{1}{2}+O(h^2)$. For the matrix D_1 associated with the C^1 cubics, $a=102/175$, $b=54/175$, giving a spectral radius of $9/26+O(h^2)$.

We now consider the enhancement of the iterative methods (3.3) in the following fashion: for both C^0 quadratic and C^1 cubic elements, solve linear systems of the form $D_1 x=y$ using (4.3). Additionally, in the case of C^1 cubic elements, solve $D_i x=y$, $i=2,3,4$, using iterative methods of the form (4.3) but employing 1-point rather than 5-point difference formulate to determine G_i.

The appropriate diagonal matrices for the C^1 cubic case are $G_2=G_3=119h^2/525I$ and $G_4=19h^4/1575I$. These iterations can be analyzed in the same fashion as that for G_1, i.e., in the appropriate transformed coordinates, $\hat{D}_i$ and $\hat{G}_i$ are diagonal. The scalar multiples in G_i were chosen such that the optimum ω was 1 in all cases. The spectral radii are $56/119+O(h^2)$ for $i=2,3$, and $16/19+O(h^2)$ for $i=4$.

Since all of the iterations occur inside the outer iteration (3.3), we do not expect to solve systems to full accuracy, but only to reduce the initial error by a factor ε which is somewhat smaller than the spectral radius of the outer iteration. Any additional accuracy would be largely wasted, since such digits are generally inaccurate in the context of the outer iteration. Thus we anticipate that a relatively small number of inner iterations will suffice.

To illustrate, we estimate the effect of inner iterations on the block SOR iteration. Let M, D, E be defined as in (3.2) and let

$$\begin{aligned}&L=D^{-1}E^T; \quad U=D^{-1}E\\ &D_i=G_i-N_i; \quad H_i=G_i^{-1}N_i; \quad 1\leqq i\leqq 4:\\ &H=\mathrm{Diag}[H_i^{\rho_i}(I-H_i^{\rho_i})^{-1}], \quad \rho_i \text{ an integer, } 1\leqq i\leqq 4.\end{aligned} \tag{4.5}$$

It is easy to verify by induction that solving $D_i v=z$ using the ρ_i step iteration

$$G_i v_k=N_i v_{k-1}+z; \quad k=1,2,\ldots,\rho_i; \quad v_0 \text{ given}, \tag{4.6}$$

is equivalent mathematically to solving the linear system

$$D_i(I-H_i^{\rho_i})^{-1}v_{\rho_i}=D_i H_i^{\rho_i}(I-H_i^{\rho_i})^{-1}v_0+z \tag{4.7}$$

once. If the current best estimate for the solution is taken as the initial guess for the inner iterations, then block SOR with inner iterations can be written as

$$[D(I+H)-\omega E](x_{k+1}-x_k)=\omega(b=Mx_k), \tag{4.8a}$$

or more conveniently, using (3.2) and (4.5)

$$(I+H-\omega L)x_{k+1}=(H+(1-\omega)I+\omega U)x_k+\omega D^{-1}b, \tag{4.8b}$$

$k=0,1,\ldots;x_0$ given.

To find the spectral radius, we look at the iteration (4.8b) in transformed coordinates. For the particular form of the inner iterations we have chosen, H, the reordered transform of H, is diagonal. In particular, for the 4×4 matrix of interest, $\overline{H}$, we have $\overline{H}=\mathrm{Diag}[\delta_i/(1-\delta_i)]$ where δ_i is the spectral radius of the i-th block raised to the ρ_i-th power. To find the overall spectral radius, we seek λ satisfying

$$\mathrm{Det}[\lambda-1)\overline{H}+(\lambda-1+\omega)I-\omega(\overline{U}+\lambda\bar{L})]=0 \tag{4.9}$$

where $\overline{H}$, $\bar{L}$, and $\overline{U}$ are the appropriate 4×4 matrices. (From Theorem 3.1, we know the solution when $\overline{H}\equiv 0$.) Now suppose $\overline{H}=\mathrm{Diag}[\delta/(1-\delta)]$, $0\leqq\delta<1$. This approximation allows the trivial solution of (4.9), yielding a qualitative picture of the actual situation. In particular, let $\tilde{\omega}=\omega(1-\delta)$; then (4.9) is equivalent to finding λ such that

$$\mathrm{Det}[(\lambda-1+\tilde{\omega})I-\tilde{\omega}(\overline{U}+\lambda\bar{L})]=0. \tag{4.10}$$

This is precisely (4.9) with $H\equiv 0$ and $\tilde{\omega}$ replacing ω. Thus (4.10) gives an optimum spectral radius which is equal to the optimum spectral radius without inner iterations, and an optimum ω which is larger by a factor of $1/(1-\delta)$. Qualitatively, we expect the effect of inner iterations to be small provided that the δ_i are sufficiently small. This has been verified by preliminary numerical experiments. These experiments also indicate that when the δ_i are large compared to the spectral radius of the outer iteration (for example, in the C^1 cubic case with $\rho_4=1$, $\delta_4=16/19+O(h^2)$), then the rate of convergence of the overall iteration is dominated by the rate of convergence of the inner iterations.

We conclude with several remarks. First, in the case of C^1 cubic elements, one could improve the performance of the inner iterations for $i=2,3,4$, by using G_i which are derived from 3- or 5-point difference formulae. The analysis of such cases can be carried out as above. For example, in the case $i=2$ or 3, an optimal 3-point difference scheme reduces the spectral radius of that inner iteration from $56/119+O(h^2)$ to $1/4+O(h^2)$. The question of selecting the type and number of inner iterations in a manner which minimizes the total cost of solving the problem is a matter for future research. Second, as was the case in Section 3, rescaling the basis functions does not affect the rate of convergence; however, some of the constants would be altered.

5. Concluding Remarks

(A) The results of Sections 2–4 can be extended in reasonably straightforward fashion to other boundary condition combinations, similar to the analogous

extensions for 5-point finite difference formulae [3, 13]. The appropriate Fourier transforms vary from case to case, as do several other details. In [16], several of the results of Sections 3–4 are extended, within the framework of "multi-level" schemes, to cover more general elliptic operators, irregular regions and meshes, and a broader class of finite element methods. For this general class of iterative schemes, the rate of convergence can be shown to be independent of h under reasonably weak hypotheses. In the more general setting, however, general sparse elimination algorithms surplant the use of fast direct methods.

(B) The spectral radii given in Table 3.1 are basis dependent. For example, suppose we use the Lagrange interpolating basis [12] for the C^0 quadratic space rather than (1.4). In the partitioning corresponding to (3.1) all the U_i would correspond to u. The spectral radii of the iterations (3.3) are not essentially independent of n in this case, and $O(n \log 1/\varepsilon)$ iterations are required to reduce the error by ε. Another open question is whether there exist other convenient bases for these subspaces, which reduce the spectral radii of the iterative methods.

(C) These methods extend readily to variable coefficient problems. For example, consider

$$\begin{aligned} &-\nabla \cdot a \nabla u = f \quad \text{in} \quad \Omega = (0,1) \times (0,1); \\ &\qquad u = 0 \quad \text{on} \quad \partial\Omega; \\ &0 < a_0 \leqq a(x,y) \leqq a_1, \qquad (x,y) \in \Omega. \end{aligned} \tag{5.1}$$

To compute an approximate solution of (5.1) using either C^0 quadratic or C^1 cubic finite elements, we must solve a linear system $Mx = b$. Let $\overline{M}$ denote the tensor product matrix corresponding to the operator $-\Delta$ on the same grid. We solve $Mx = b$ using

$$\overline{M}(x_{k+1} - x_k) = \omega(b - Mx_k); \qquad x_0 \text{ given}, \quad k = 0, 1, \ldots. \tag{5.2}$$

It is convenient to analyze the convergence of the iteration (5.2) using

Theorem 5.1. Let M and $\overline{M}$ be symmetric, real, and positive definite. Let μ_1 and μ_2 be positive numbers such that, for all $x \neq 0$ we have

$$x^T M x / x^T \overline{M} x \in [\mu_1, \mu_2].$$

Then for $0 < \omega < 2/\mu_2$ the sequence x_k defined in (5.2) converges to $M^{-1} b$. Further, for $\omega = 2/(\mu_1 + \mu_2)$, the M-norm of the error is reduced by a factor of at least $(\mu_2 - \mu_1)/(\mu_2 + \mu_1)$ in each iteration. (The M-norm of a vector x is given by $\|x\|_M = \|M^{1/2} x\|_2 = (x^T M x)^{1/2}$.)

Theorem 5.1 is proved in [10] among others. For the finite element matrices

$$v^T M v = a(\phi, \phi) = \int_\Omega a \nabla \phi \, \nabla \phi \, dx \tag{5.3}$$

where ϕ is an element of the finite dimensional subspace, and is characterized by the coefficient vector v. Since

$$v^T \overline{M} v = \int_\Omega \nabla \phi \, \nabla \phi \, dx \tag{5.4}$$

it follows that

$$a_0 \leqq \frac{v^T M v}{v^T \overline{M} v} \leqq a_1 \tag{5.5}$$

for all $v \neq 0$. Thus (5.2) converges with spectral radius bounded independent of n. This spectral radius can often be substantially reduced by an appropriate change of variables in the original problem [2, 4]. It is also advantageous to apply some acceleration procedure to (5.2), for example, Chebyshev [4], or preconditioned conjugate gradient [5, 8]. In either event, effective methods for solving $\overline{M} x = b$ play a central role.

References

1. Bank, R.E., Rose, D.J.: Marching algorithms for elliptic boundary value problems. I. The constant coefficient case. SIAM J. Numer. Anal. **14**, 792–829 (1977)
2. Bank, R.E.: Marching algorithms for elliptic boundary value problems. II. The variable coefficient case. SIAM J. Numer. Anal. **14**, 950–970 (1977)
3. Buzbee, B.L., Golup, G.H., Neilson, C.W.: On direct methods for solving Poisson's equation. SIAM J. Numer. Anal. **7**, 627–656 (1970)
4. Concus, P., Golup, G.H.: The use of fact direct methods for the efficient solution of non-separable elliptic equations. SIAM J. Numer. Anal. **10**, 1103–1120 (1973)
5. Concus, P., Golup, G.H., O'Leary, D.P.: A generalized conjugate iteration for the numerical solution of elliptic partial differential equations. In: Sparse matrix computations (J.R. Bunch, D.J. Rose, eds.). New York-London: Academic Press 1976
6. Cooley, J.W., Lewis, P.A.W., Welch, P.D.: The fast Fourier transform algorithm: programming considerations in the calculations of sine, cosine and Laplace transforms. J. Sound Vib. **12**, 315–337 (1970)
7. Dorr, F.W.: The direct solution of the discrete Poisson equation on a rectangle. SIAM Rev. **12**, 248–263 (1970)
8. Douglas, J. Jr., Dupont, T.: Preconditioned conjugate gradient iteration applied to Galerkin methods for midly nonlinear Dirichlet problem. In: Sparse matrix computations (J.R. Bunch, D.J. Rose, eds.). New York-London: Academic Press 1976
9. Forsythe, G.E., Moler, C.B.: Computer solution of linear algebraic systems. Prentice Hall 1967
10. Gunn, J.E.: The solution of difference equations by semi-explicit iterative techniques. SIAM J. Numer. Anal. **2**, 24–45 (1965)
11. Schultz, M.H.: Spline analysis. Prentice Hall 1973
12. Strang, G., Fix, G.: An analysis of the finite element method. Prentice Hall 1973
13. Swarztrauber, P.N.: The methods of cyclic reduction, Fourier analysis, and cyclic reduction – Fourier analysis in the discrete solution of Poisson's equation on a rectangle. SIAM Rev. (to appear)
14. Varga, R.S.: Matrix iterative analysis. Prentice Hall 1962
15. Wilkinson, J.H.: The algebraic eigenvalue problem. The Clarendon Press 1965
16. Bank, R.E., Dupont, T.: An optimal order process for solving finite element equations. Math. Comput. (submitted)

Received May 10, 1977