A New Iterative Chebyshev Spectral Method for Solving the Elliptic Equation $\nabla \cdot (\sigma \nabla u) = f$

SHENGKAI ZHAO AND MATTHEW J. YEDLIN

Department of Geophysics and Astronomy, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z4

Received November 4, 1992; revised October 28, 1993

We present a new iterative Chebyshev spectral method for solving the elliptic equation $\nabla \cdot (\sigma \nabla u) = f$. We rewrite the equation in the form of a Poisson's equation $\nabla^2 u = (f - \nabla u \cdot \nabla \sigma)/\sigma$. In each iteration we compute the right-hand side terms from the guess values first. Then we solve the resultant Poisson equation by a direct method to obtain the updated values. Three numerical examples are presented. For the same number of iterations, the accuracy of the present method is about 6–8 orders better than the Chebyshev spectral multigrid method. On a SPARC Station 2 computer, the CPU time of the new method is about one-third of the Chebyshev spectral multigrid method. To obtain the same accuracy, the CPU time of the present method is about one-tenth of the Chebyshev spectral multigrid method. © 1994 Academic Press, Inc.

1. INTRODUCTION

There have been many papers describing the solution of elliptic equations by Chebyshev spectral methods. Zang *et al.* [12, 13] solved the 2D elliptic problem with spectral multigrid methods. They used an incomplete LU decomposition technique to solve the finite difference (FD) preconditioning equations and gave several examples. Brandt, Fulton, and Taylor [1] proposed several modifications to the methods presented by Zang *et al.* for periodic problems. Deville and Mund [3, 4] discussed the Chebyshev pseudospectral algorithm for second-order elliptic equations using finite element (FE) preconditioning. They computed the spectral radius of the elliptic solvers with FD and FE preconditioning and for different interpolation elements. Canuto *et al.* [2, Chap. 5] and Zang *et al.* [14] gave comprehensive discussions and reviews of the Chebyshev spectral methods in solving elliptic equations. In this paper we present a new iterative Chebyshev spectral method for solving the elliptic equation $\nabla \cdot (\sigma \nabla u) = f$. In Section 2, we present the main idea of the new method. In Section 3, we discuss how to solve the Poisson's equation by the Chebyshev collocation method. In Section 4 we present the formulation of the new method for 1D and 2D elliptic equations. In Section 5, we present numerical examples. In Section 6, we discuss the results.

2. A NEW ITERATION METHOD FOR SOLVING THE ELLIPTIC EQUATION $\nabla \cdot (\sigma \nabla u) = f$

We now consider the elliptic problem

$$\nabla \cdot (\sigma \nabla u) = f \qquad \text{in } \Omega, \tag{1a}$$

$$u = g \qquad \text{on } \Gamma, \tag{1b}$$

where Ω is the domain, Γ is the boundary, f and g are known functions, and $\sigma \geqslant a > 0$. In this paper all the problems will be considered in a standard Chebyshev domain, that is,

$$\Omega = \begin{cases} \{x \mid -1 < x < 1\} & \text{for 1D problems,} \\ \{(x, y) \mid -1 < x, y < 1\} & \text{for 2D problems,} \\ \{(x, y, z) \mid -1 < x, y, z < 1\} & \text{for 3D problems.} \end{cases}$$

If $\nabla \sigma$ exists in the whole domain, we can rewrite Eq. (1a) in the form of a canonical Poisson's equation

$$\nabla^2 u = \frac{f - \nabla \sigma \cdot \nabla u}{\sigma}. \tag{2}$$

Then we solve the problem by the following procedure:

1. For a given guess of u, compute the right-hand side of (2).
2. Solve the resultant Poisson's equation by the direct method to be discussed in the next section.
3. Take the solution obtained as new guess and repeat the above process until the norms of the differences of u between two successive iterations satisfy an appropriate convergence criterion.

Since σ is given, $\nabla \sigma$ can be computed analytically or numerically. However, ∇u has to be computed numerically.

0021-9991/94 $6.00

3. SOLVING THE POISSON'S EQUATION BY CHEBYSHEV COLLOCATION METHODS

With the new method we need to solve a Poisson problem in each iteration. In this section, we discuss how to solve the Poisson's problem

$$\nabla^2 u = h \qquad \text{in } \Omega, \tag{3a}$$

$$u = \text{known} \qquad \text{on } \Gamma, \tag{3b}$$

by the Chebyshev collocation method, where h is a known function. There are many papers describing how to solve the Poisson's equation by the Chebyshev spectral methods. Haidvogel and Zang [16] used the τ method to solve the 2D Poisson's equation in a square with homogeneous boundary conditions and proposed both alternating-direction implicit iteration and matrix-diagonalization methods to solve the coefficient equation system. Haldenwang *et al.* [17] extended Haidvogel and Zang's 2D method to the 3D Helmholtz equation with general inhomogeneous boundary conditions. In the τ method, the equation system is derived for the Chebyshev coefficients. In order to build the equation system we have to apply a forward Chebyshev transform to h to convert the quantity from physical space to transformed space. After obtaining the Chebyshev coefficients of u, we have to apply an inverse Chebyshev transform to convert the quantity from transformed space to physical space. In the collocation method, we require that Eq. (3a) be satisfied exactly at the interior points and solve the equation system in physical space directly. No Chebyshev transforms are required. Therefore, it is simpler and faster than the τ method. Besides, in the τ method we need to know the values of h at the boundaries (which are not required in the collocation method) in order to obtain the Chebyshev coefficients of the source function. This makes the problem somewhat overdetermined. When the boundary conditions and the right-hand side terms are self-consistent, the τ method works well. When the boundary conditions are given arbitrarily (as happened when using the influence matrix method to solve the multidomain equation system [15]), because of a lack of consistency, the τ method does not work. In contrast, the collocation method works well because it is not overdetermined. Furthermore, the collocation method can be used to solve more general equations. For simplicity, we use the same degree N of Chebyshev polynomials in each coordinate direction. The Chebyshev grid is

$$x_i = y_i = z_i = \cos\frac{i\pi}{N} \qquad (i = 0, ..., N). \tag{4}$$

We will discuss 1D problem in detail to illustrate the main ideas and then generalize them to 2D and 3D problems.

3.1. *The First and Second Derivative Matrices*

Let

$$\bar{C}_i = \begin{cases} 1 & \text{for} \quad i = 1, ..., N-1, \\ 2 & \text{for} \quad i = 0 \quad \text{or} \quad N. \end{cases} \tag{5}$$

The first-order derivative matrix $[d^{(1)}_{ij}]$ is given by [13]

$$d^{(1)}_{ij} = \begin{cases} \dfrac{\bar{C}_i(-1)^{i+j}}{\bar{C}_j(x_i - x_j)}, & 0 \leqslant i, j \leqslant N, \quad j \neq i, \\ -\dfrac{x_j}{2(1-x_j^2)}, & 1 \leqslant i = j \leqslant N-1, \\ \dfrac{2N^2+1}{6}, & i = j = 0, \\ -\dfrac{2N^2+1}{6}, & i = j = N. \end{cases} \tag{6}$$

The second-order derivative matrix $[d^{(2)}_{ij}]$ is given by [8]

$$d^{(2)}_{ij} = \begin{cases} \dfrac{(-1)^{i+j}(x_i^2 + x_i x_j - 2)}{\bar{C}_j(1-x_i^2)(x_i - x_j)^2}, \\ \qquad 1 \leqslant i \leqslant N-1, \quad 0 \leqslant j \leqslant N; \quad j \neq i, \\ -\dfrac{(N^2-1)(1-x_i^2)+3}{3(1-x_i^2)^2}, \\ \qquad 1 \leqslant i = j \leqslant N-1, \\ \dfrac{(-1)^j 2[(2N^2+1)(1-x_j)-6]}{3\bar{C}_j(1-x_j)^2}, \\ \qquad i = 0, \quad 1 \leqslant j \leqslant N, \\ \dfrac{(-1)^{N+j} 2[(2N^2+1)(1+x_j)-6]}{3\bar{C}_j(1+x_j)^2}, \\ \qquad i = N, \quad 0 \leqslant j \leqslant N-1, \\ \dfrac{N^4-1}{15}, \\ \qquad k = j = 0, N. \end{cases} \tag{7}$$

It is easy to verify that

$$d^{(2)}_{(N-i)(N-j)} = d^{(2)}_{ij}. \tag{8}$$

If the discrete values $u_l = u(x_l)$ are given, the first- and the second-order derivatives of $u(x)$ at the grid points can be computed by

$$u_i^{(1)} = \sum_{l=0}^{N} d^{(1)}_{il} u_l, \tag{9}$$

$$u_i^{(2)} = \sum_{l=0}^{N} d^{(2)}_{il} u_l. \tag{10}$$

Formulas (9) and (10) can be easily extended to 2D and 3D cases.

3.2. *The 1D Problem*

In the 1D case, Eq. (3a) becomes

$$\frac{d^2u}{dx^2} = h(x). \tag{11}$$

From (7), (10), and (11), we obtain the equations

$$\sum_{l=0}^{N} d_{il}^{(2)} u_l = h_i \qquad (i = 1, ..., N-1). \tag{12}$$

From (3b) u_0 and u_N are known. Moving the known terms on the left-hand side of (12) to the right-hand side, we obtain

$$\sum_{l=1}^{N-1} \tilde{d}_{il}^{(2)} u_l = s_i \qquad (i = 1, ..., N-1), \tag{13}$$

where

$$\tilde{d}_{il}^{(2)} = d_{il}^{(2)} \qquad (i, l = 1, ..., N-1) \tag{14}$$

and

$$s_i = h_i - d_{i0}^{(2)} u_0 - d_{iN}^{(2)} u_N. \tag{15}$$

System (13) can be solved by many different methods such as Gauss elimination or Gauss–Seidel iteration. For a better understanding of the 2D and 3D problems to follow, we solve system (13) by the matrix diagonalization method. The matrix $[\tilde{d}_{ij}^{(2)}]$ can be factorized as

$$\tilde{d}_{il}^{(2)} = \sum_{j=1}^{N-1} \alpha_{ij} \lambda_j \alpha_{jl}^{-1} \qquad (1 \leqslant i, j \leqslant N-1), \tag{16}$$

where $[\alpha_{ij}]$ is an $(N-1)(N-1)$ matrix whose ith column is the eigenvector corresponding to the ith eigenvalue λ_i of matrix $[\tilde{d}_{ij}^{(2)}]$, and $[\alpha_{jl}^{-1}]$ is the inverse of $[\alpha_{ij}]$. Therefore,

$$\sum_{j=1}^{N-1} \alpha_{ij} \alpha_{jl}^{-1} = \sum_{j=1}^{N-1} \alpha_{ij}^{-1} \alpha_{jl} = \delta_{il} = \begin{cases} 1, & i = l \\ 0, & i \neq l. \end{cases} \tag{17}$$

All the eigenvalues are real, negative, and distinct [6]. The condition number of the matrix $[\tilde{d}_{il}^{(2)}]$ is $O(N^4)$. Because of (8) we have

$$\alpha_{i(N-j)} = (-1)^{N+j+1} \alpha_{ij} \tag{18}$$

and

$$\alpha_{(N-i)j}^{-1} = (-1)^{N+i+1} \alpha_{ij}^{-1}. \tag{19}$$

We can use (18) and (19) to reduce the amount of computation. Substituting (16) into (13), we obtain

$$\sum_{l=1}^{N-1} \sum_{j=1}^{N-1} \alpha_{ij} \lambda_j \alpha_{jl}^{-1} u_l = s_i \qquad (i = 1, 1, ..., N-1). \tag{20}$$

We then multiply the ith equation of (20) by α_{mi}^{-1} and sum over i to obtain

$$\sum_{i=1}^{N-1} \alpha_{mi}^{-1} \sum_{l=1}^{N-1} \sum_{j=1}^{N-1} \alpha_{ij} \lambda_j \alpha_{jl}^{-1} u_l = \sum_{i=1}^{N-1} \alpha_{mi}^{-1} s_i. \tag{21}$$

Let

$$w_j = \sum_{l=1}^{N-1} \alpha_{jl}^{-1} u_l \tag{22}$$

and

$$t_m = \sum_{i=1}^{N-1} \alpha_{mi}^{-1} s_i. \tag{23}$$

From (17), (21), (22), and (23), we obtain

$$\sum_{j=1}^{N-1} \delta_{mj} \lambda_j w_j = t_m. \tag{24}$$

Therefore,

$$w_m = t_m / \lambda_m. \tag{25}$$

From (22)

$$u_j = \sum_{m=1}^{N-1} \alpha_{jm} w_m. \tag{26}$$

The above solution process can be summarized in two steps:

$$w_i = \frac{1}{\lambda_i} \sum_{l=1}^{N-1} \alpha_{il}^{-1} s_l \qquad (i = 1, ..., N-1), \tag{27}$$

$$u_l = \sum_{i=0}^{N-1} \alpha_{li} w_i \qquad (l = 1, ..., N-1). \tag{28}$$

As pointed out by Canuto *et al.* [2, p. 135], this strategy is an application of the tensor product approach devised by Lynch, Rice, and Thomas [7] for finite-difference approximation to Poisson's equation. The above algorithm can be extended to 2D and 3D problems easily. Because the 3D formulation is very similar to the 2D, we present the 2D formulation only.

3.3. *The 2D Problem*

In the 2D case, Eq. (3a) becomes

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = h(x, y). \tag{29}$$

Analogous to the 1D case, we obtain the equation system

$$\sum_{l=1}^{N-1} \tilde{d}_{il}^{(2)} u_{lj} + \sum_{l=1}^{N-1} \tilde{d}_{jl}^{(2)} u_{il} = h_{ij} \qquad (i, j = 1, ..., N-1), \tag{30}$$

where

$$s_{ij} = h_{ij} - d_{i0}^{(2)} u_{0j} - d_{iN}^{(2)} u_{Nj} - d_{j0}^{(2)} u_{i0} - d_{jN}^{(2)} u_{iN}.$$

We solve system (30) using the following two steps:

$$w_{ij} = \frac{1}{\lambda_i + \lambda_j} \sum_{l=1}^{N-1} \sum_{m=1}^{N-1} \alpha_{il}^{-1} \alpha_{jm}^{-1} s_{lm} \qquad (i, j = 1, ..., N-1), \tag{31}$$

$$u_{lm} = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \alpha_{li} \alpha_{mj} w_{ij} \qquad (l, m = 1, ..., N-1). \tag{32}$$

4. SOLVING THE ELLIPTIC EQUATION

In this section, we present the formulation for solving the elliptic boundary value problem (1) in 1D and 2D cases. As a preface, we briefly introduce the spectral radius and convergence rate. Then we estimate them for the 1D problem.

4.1. *The Iteration Matrix and Its Spectral Radius*

Let $[g_{ij}]$ be an $N \times N$ matrix with eigenvalues λ_i $(1 \leqslant i \leqslant N)$. The spectral radius of the matrix $[g_{ij}]$ is defined as [10]

$$\rho([g_{ij}]) = \max_{1 \leqslant i \leqslant N} |\lambda_i|. \tag{33}$$

Suppose we solve the equation

$$u_i = s_i + \sum_{r=1}^{N} g_{ir} u_r \qquad (i = 1, ..., N) \tag{34}$$

by the iteration relation

$$u_i^{(k+1)} = s_i + \sum_{r=1}^{N} g_{ir} u_r^{(k)}. \tag{35}$$

Subtracting (34) from (35) we obtain

$$u_i^{(k+1)} - u_i = \sum_{r=1}^{N} g_{ir} (u_r^{(k)} - u_r). \tag{36}$$

Equation (36) implies that the error $e_i^{(k)} = u_i^{(k)} - u_i$ obeys

$$e_i^{(k+1)} = \sum_{r=1}^{N} g_{ir} e_r^{(k)}. \tag{37}$$

Suppose that

$$e_i^{(0)} = \sum_{r=1}^{N} c_r (\hat{e}_r)_i, \tag{38}$$

where $(\hat{e}_r)_i$ is the ith element of the rth eigenvector of matrix $[g_{ij}]$ and the c_r's are constants. Then after k iterations the error is given by

$$e_i^{(k)} = \sum_{p=1}^{N} g_{ip} e_p^{(k-1)} = \cdots = \sum_{r=1}^{N} c_r \lambda_r^k (\hat{e}_r)_i. \tag{39}$$

When k becomes large enough, $e_i^{(k)}$ will be dominated by the largest eigenvalue(s). That is, the error $e_i^{(k)}$ is determined by the spectral radius of the iteration matrix. The convergence rate $\mathcal{R}$ is defined as [2, p. 138]

$$\mathcal{R} = -\ln \rho. \tag{40}$$

The reciprocal of $\mathcal{R}$ measures the number of iterations required to reduce the error by a factor of e. The larger the convergence rate is the fewer iterations that are required to obtain a solution to a given accuracy.

4.2. *The 1D Problem*

In the 1D case, Eq. (1a) becomes

$$\frac{d}{dx}\left(\sigma \frac{du}{dx}\right) = f \qquad \text{in } \Omega. \tag{41}$$

From (2), (9), and (10), the discrete form of (41) is

$$\sum_{j=0}^{N} d_{ij}^{(2)} u_j = \frac{f_i}{\sigma_i} - \frac{\sigma_i'}{\sigma_i} \sum_{j=0}^{N} d_{ij}^{(1)} u_j \qquad (i = 1, ..., N-1). \tag{42}$$

After moving the terms containing the known boundary values, u_0 and u_N, to the left-hand side of Eq. (42), we obtain the final equation system,

$$\sum_{j=1}^{N-1} \tilde{d}_{ij}^{(2)} u_j = s_i - \frac{\sigma_i'}{\sigma_i} \sum_{j=1}^{N-1} \tilde{d}_{ij}^{(1)} u_j \qquad (i = 1, ..., N-1), \tag{43}$$

where

$$s_i = \frac{f_i}{\sigma_i} - \frac{\sigma_i'}{\sigma_i} (d_{i0}^{(1)} u_0 + d_{iN}^{(1)} u_N) - d_{i0}^{(2)} u_0 - d_{iN}^{(2)} u_N, \tag{44}$$

$$\tilde{d}_{il}^{(1)} = d_{il}^{(1)} \qquad (i, j = 1, ..., N-1), \tag{45}$$

and $d_{il}^{(1)}$ is defined by (6). From (43) the iteration formula for the present method is

$$u_i^{(k+1)} = c_i - \sum_{j=1}^{N-1} g_{ij} u_j^{(k)}, \tag{46}$$

where

$$c_i = \sum_{r=1}^{N-1} [\tilde{d}_{ir}^{(2)}]^{-1} s_r,$$

and the iteration matrix is

$$g_{ij} = -\sum_{r=1}^{N-1} [\tilde{d}_{ir}^{(2)}]^{-1} \frac{\sigma'_r}{\sigma_r} \tilde{d}_{rj}^{(1)}. \tag{47}$$

From (47) we can find the spectral radius of $[g_{ij}]$. The iteration matrix is model-dependent, as is the spectral radius. We can compute the spectral radius for 2D and 3D problems in a similar way.

Now we give an estimate of the convergence rate of the 1D problem. We exchange the order of the matrices on the right-hand side of (47) to obtain another matrix

$$q_{ij} = -\sum_{k=1}^{N-1} \frac{\sigma_i^{(x)}}{\sigma_i} \tilde{d}_{ik}^{(1)} [\tilde{d}^{(2)}]_{kj}^{-1}. \tag{48}$$

The matrices $[g_{ij}]$ and $[q_{ij}]$ have the same eigenvalues [11, p. 54] and therefore the same spectral radii. Let

$$b_{ij} = \sum_{k=1}^{N-1} \tilde{d}_{ik}^{(1)} [\tilde{d}^{(2)}]_{kj}^{-1}.$$

Then

$$q_{ij} = -\frac{\sigma_i^{(x)}}{\sigma_i} b_{ij}.$$

Numerical experiments show that the spectral radius of the matrix $[b_{ij}]$ is bounded by $1/\pi$. Since the matrix $[\sigma_i^{(x)}/\sigma_i]$ is diagonal, the spectral radius is the maximum absolute value of its elements. From [10, p. 5], we have

$$\rho([q_{ij}]) \leqslant \rho\left(\left[\frac{\sigma_i^{(x)}}{\sigma_i}\right]\right) \rho([b_{ij}]) \leqslant \frac{1}{\pi} \max\left(\frac{d \ln \sigma}{dx}\right).$$

Therefore, the spectral radius of the iteration matrix $[g_{ij}]$ is

$$\rho([g_{ij}]) \leqslant \frac{1}{\pi} \max\left|\frac{d \ln \sigma}{dx}\right|. \tag{49}$$

From (40) and (49), the convergence rate $\mathscr{R}$ for the 1D problem is bounded by

$$\mathscr{R} \geqslant \ln \pi - \ln\left(\max\left|\frac{d \ln \sigma}{dx}\right|\right). \tag{50}$$

4.3. *The 2D Problem*

In the 2D case, Eq. (1a) becomes

$$\frac{\partial}{\partial x}\left(\sigma \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\sigma \frac{\partial u}{\partial y}\right) = f(x, y) \qquad \text{in } \Omega, \tag{51}$$

where $\Omega = \{(x, y) \mid \bar{x}_2 < x < \bar{x}_1,\ \bar{y}_2 < y < \bar{y}_1\}$. We rewrite Eq. (51) as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{f - u^{(x)}\sigma^{(x)} - u^{(y)}\sigma^{(y)}}{\sigma}. \tag{52}$$

Similar to the derivation for the 1D problem in the previous section, from (3), (8), and (9), we obtain a system of equations

$$\begin{aligned} &\sum_{l=1}^{N-1} \tilde{d}_{il}^{(2)} u_{lj} + \sum_{l=1}^{N-1} \tilde{d}_{jl}^{(2)} u_{il} \\ &\quad = s_{ij} - \frac{\sigma_{ij}^{(x)}}{\sigma_{ij}} \sum_{l=1}^{N-1} \tilde{d}_{il}^{(1)} u_{lj} - \frac{\sigma_{ij}^{(y)}}{\sigma_{ij}} \sum_{l=1}^{N-1} \tilde{d}_{jl}^{(1)} u_{il} \\ &\quad\quad (0 < i,\ j < N), \end{aligned} \tag{53}$$

where

$$\begin{aligned} s_{ij} = {} & \frac{1}{\sigma_{ij}} [f_{ij} - \sigma_{ij}^{(x)}(d_{i0}^{(1)} u_{0j} + d_{iN}^{(1)} u_{Nj}) \\ & - \sigma_{ij}^{(y)}(d_{j0}^{(1)} u_{i0} + d_{jN}^{(1)} u_{iN})] \\ & - (d_{i0}^{(2)} u_{0j} + d_{iN}^{(2)} u_{Nj}) - (d_{j0}^{(2)} u_{i0} + d_{jN}^{(2)} u_{iN}). \end{aligned} \tag{54}$$

The formulation for the 3D problem is similar to the 2D case and it omitted for brevity.

5. NUMERICAL EXAMPLES

Three examples are now presented. In each example we compute the solution by both the present method and the spectral multigrid method presented by Zang *et al.* [12]. In the multigrid method, a simple V cycle with one relaxation on each grid level is used. The correction vector is obtained by solving a finite difference preconditioning equation system.

The convergence rate of any iteration process is characterized by the L_∞ or L_2 norms of the difference of the values of the unknowns between two successive iterations. In the following examples, we will plot the L_∞ norms.

The accuracy of the approximation (compared with the exact solution) is determined by the degree of Chebyshev polynomials used as well as the number of iterations. For comparison we also present the error norms and CPU time after a number of iterations for both the new iteration method and the multigrid method presented by Zang *et al.* for each example.

All the computations were performed on a SPARC Station 2 computer with double precision.

EXAMPLE 1. In Eq. (1a), we take

$$\sigma(x) = 1 + \varepsilon x^2,$$

for the exact solution

$$u(x) = \cos(x^2).$$

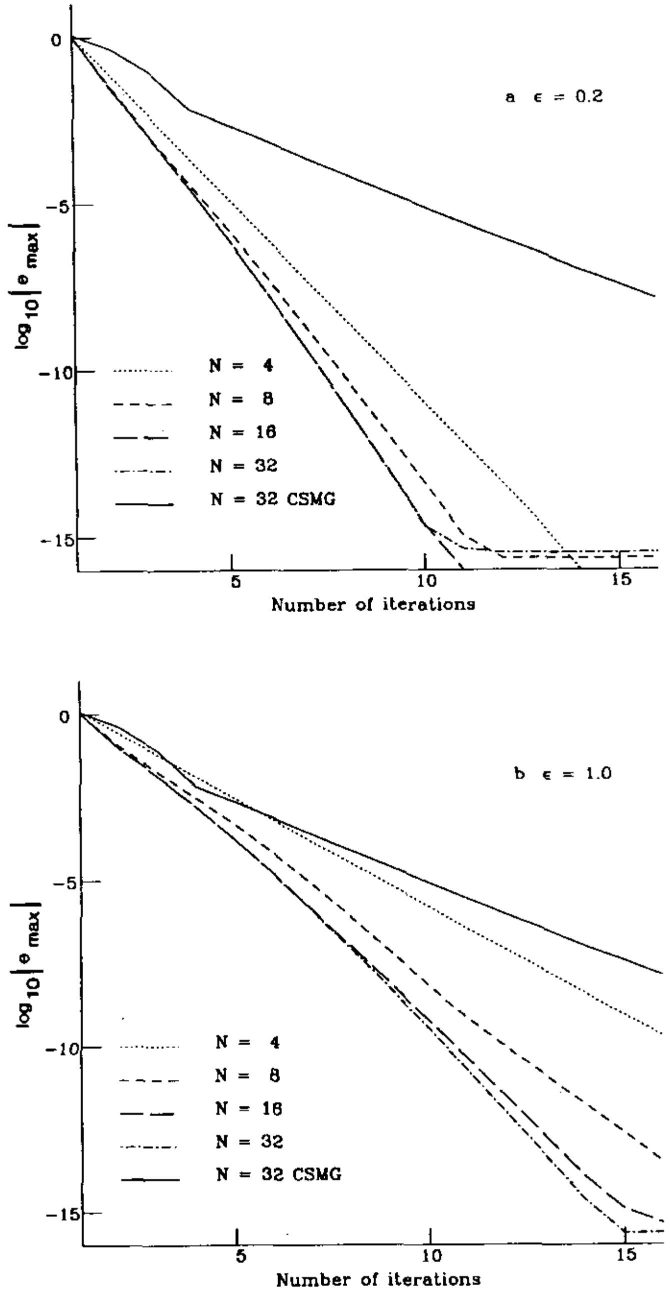

FIG. 1. The L_∞ norms of the differences of u between two successive iterations of Example 1: (a) $\varepsilon = 0.2$; (b) $\varepsilon = 1$.

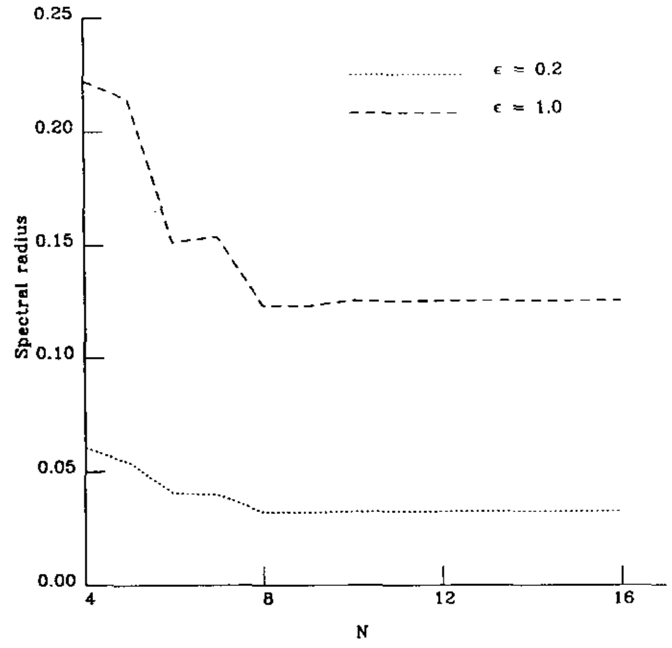

FIG. 2. The spectral radius of the iteration matrix for Example 1.

The boundary conditions and $f(x)$ are given accordingly. Figure 1 shows the L_∞ norms ($|e_{\max}|$) of the differences of u between two successive iterations for the cases $N = 4$, 8, 16, and 32 for the present method and that for Chebyshev spectral multigrid methods (CSMG) on the $N = 32$ grid. In Fig. 1, (a) is for $\varepsilon = 0.2$ and (b) is for $\varepsilon = 1$. Figure 2 shows the spectral radius for $N = 4$ to 16. Table I shows the L_2 and L_∞ error norms (comparing with the exact solutions) and the CPU time after nine iterations for both the present method and the spectral multigrid methods.

For this 1D example, the maximum absolute value of $d \ln \sigma/dx$ is $\sqrt{\varepsilon}$. From (50), the estimated convergence rate of the iteration matrix $[g_{ij}]$ is $\mathscr{R}^{(e)} \geqslant \ln \pi - 0.5 \ln \varepsilon$. Therefore $R^{(e)} \geqslant 1.9494$ for $\varepsilon = 0.2$ and $R^{(e)} \geqslant 1.1447$ for $\varepsilon = 1$. From Fig. 5.2, the asymptotic value of the spectral radius of the iteration matrix is 0.0325 for $\varepsilon = 0.2$ and 0.125 for $\varepsilon = 1$. The real convergence rate is $\mathscr{R} = 3.4265$ for $\varepsilon = 0.2$ and $\mathscr{R} = 2.0794$ for $\varepsilon = 1$. The real values of the convergence rate are larger than the estimated bound. This can be explained as follows. The estimation (50) is based on an inequality

TABLE I

The L_2 and L_∞ Error Norms of Example 1 after Nine Iterations on a $N = 32$ Grid

	ε	Present method	Spectral multigrid method
L_2	0.2	6.793×10^{-13}	6.129×10^{-06}
L_∞	0.2	2.761×10^{-13}	1.250×10^{-05}
L_2	1.0	3.194×10^{-10}	1.047×10^{-04}
L_∞	1.0	2.728×10^{-10}	1.958×10^{-04}
CPU (s)		0.08	0.11

condition. If $d \ln \sigma/dx$ is a constant, the equality sign is valid in (50). Only in this situation is the estimate accurate.

Example 2. In Eq. (1a) we take

$$\sigma(x, y) = 1 + \varepsilon(x^2 + y^2)$$

for the exact solution,

$$u(x, y) = \sin(\pi \cos x) \sin(\pi \cos y).$$

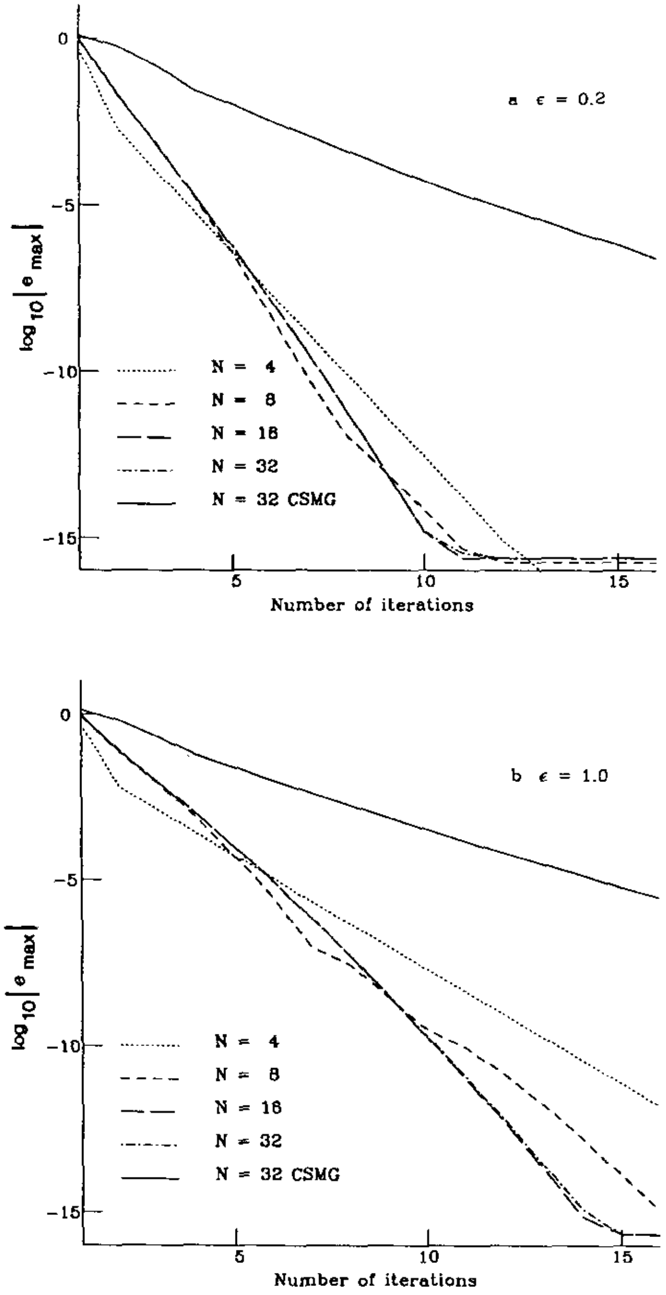

FIG. 3. The L_∞ norms of the differences of u between two successive iterations of Example 2: (a) $\varepsilon = 0.2$; (b) $\varepsilon = 1$.

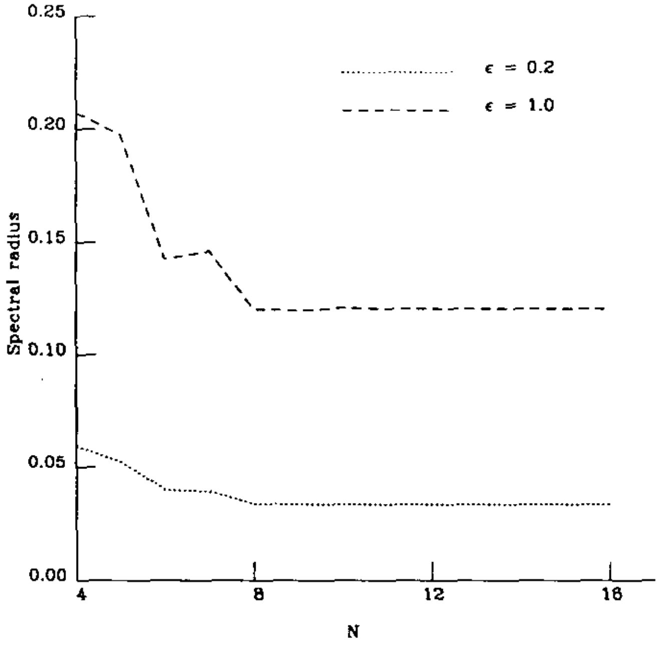

FIG. 4. The spectral radius of Example 2.

The boundary conditions and $f(x, y)$ are given accordingly. Zang *et al.* [12] solved this example by the spectral multigrid methods. Figure 3 shows the L_∞ norms of the differences of u between two successive iterations for the cases $N = 4$, 8, 16, and 32 for the present method and that for the spectral multigrid methods on the $N = 32$ grid. In Fig. 3, (a) is for $\varepsilon = 0.2$ and (b) is for $\varepsilon = 1$. Figure 4 shows the spectral radius for $N = 4$ to 16. Table II shows the L_2 and L_∞ error norms (comparing with the exact solutions) after nine iterations as well as the CPU time. Probably due to the difference in programming and the selection of relaxation parameters, in Table II the L_2 error norms (1.529×10^{-4}) for the multigrid methods is better than that (3.16×10^{-2}) presented in [12, p. 499].

Example 3. In Eq. (1a) we take

$$\sigma(x, y, z) = 1 + \varepsilon(x^2 + y^2 + z^2)$$

TABLE II

The L_2 and L_∞ Error Norms of Example 2 after Nine Iterations on a 33 × 33 Grid

	ε	Present method	Spectral multigrid method
L_2	0.2	2.816×10^{-13}	1.708×10^{-05}
L_∞	0.2	1.525×10^{-12}	8.399×10^{-05}
L_2	1.0	1.329×10^{-10}	1.172×10^{-04}
L_∞	1.0	1.938×10^{-10}	6.113×10^{-04}
CPU (s)		0.68	2.28

for the exact solution,

$$u(x, y, z) = \sin(x + y^2 + z^3).$$

The boundary conditions and $f(x, y, z)$ are given accordingly. Figure 5 shows the L_∞ norms of the differences of u between two successive iterations for the cases $N = 4$, 8, 16, and 32 for the present method and that for the spectral multigrid methods on the $N = 32$ grid. In Fig. 5, (a) is for $\varepsilon = 0.2$ and (b) is for $\varepsilon = 1$. Table III shows the L_2 and L_∞ error norms (comparing with the exact solutions) after nine iterations and the CPU time.

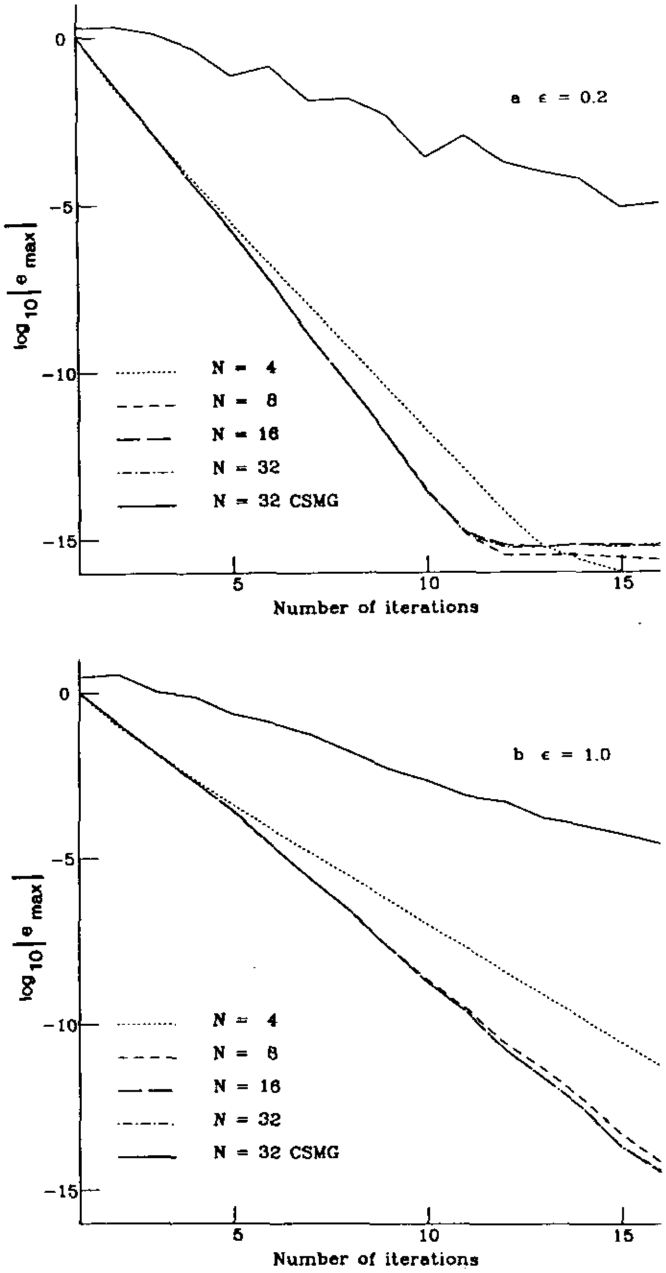

FIG. 5. The L_∞ norms of the differences of u between two successive iterations of Example 3: (a) $\varepsilon = 0.2$; (b) $\varepsilon = 1$.

TABLE III

The L_2 and L_∞ Error Norms of Example 3 after Nine Iterations on a 33 × 33 × 33 Grid

	ε	Present method	Spectral multigrid method
L_2	0.2	3.584×10^{-13}	1.850×10^{-05}
L_∞	0.2	1.422×10^{-12}	9.425×10^{-04}
L_2	1.0	4.115×10^{-10}	4.032×10^{-04}
L_∞	1.0	1.819×10^{-09}	2.580×10^{-03}
CPU (s)		35.02	157.81

6. DISCUSSION

6.1. *The Convergence Rate*

From Figs. 2 and 4 we see that for small N (<8) as N increases, the spectral radius decreases. When $N > 8$, the spectral radius essentially reaches an asymptotic value. For the 1D problem, Example 1, the asymptotic spectral radius is 0.0325 for $\varepsilon = 0.2$ and 0.125 for $\varepsilon = 1$. For the 2D problem, Example 2, the asymptotic spectral radius is 0.0335 for $\varepsilon = 0.2$ and 0.120 for $\varepsilon = 1$. We have not computed the spectral radius for the 3D problem. From Figs. 1–5 we expect that the spectral radius for the 3D problem will be slightly smaller than that for the 2D problem. From Figs. 1, 3, and 5 we see that, unlike most iterative methods, the convergence rate of the present method does not deteriorate as N increases. This is consistent with the spectral radius discussion above. When the dimension of the problem increases, the convergence rate improves slightly.

In Examples 1–3 the constant ε measures the departure of the equations from the strictly Poisson's form. The convergence rate depends on the values of ε. If $\varepsilon = 0$, the problems become the exact Poisson's equation. Thus, no iterations are required. When ε increases the convergence rate deteriorates. This can be seen by comparing the results for $\varepsilon = 0.2$ and $\varepsilon = 1.0$ in Figs. 1–5 and Tables I–III.

In Figs. 1, 3, and 5, the convergence curves for $N = 16$ and $N = 32$ are almost identical. This is due to the almost identical spectral radius. When $N = 4$ all the eigenvalues of the iteration matrix are real. In all three examples the convergence curves for $N = 4$ are nearly straight lines. The wave-like convergence curves for $N = 8$ are presumably due to the dominant complex eigenvalues of the iteration matrix. When $N = 16$ and 32, because there are many closely distributed complex eigenvalues, this phenomena is not as obvious as the $N = 8$ cases.

6.2. *Comparison with the Spectral Multigrid Method*

In Figs. 1, 3, and 5, the solid lines on the top show the L_∞ norms of the differences of u between two successive iterations obtained by the spectral multigrid methods presented by Zang *et al.* [12] on the $N = 32$ grid. We see that the convergence rate of the present method is much better than that of the spectral multigrid method. In Tables I–III, after nine iterations the error norms of the present method are about 5–8 orders better than those of the spectral multigrid method. The speed of the present method is also superior to the spectral multigrid method. For 2D and 3D problems, the CPU time of the present method is less than one-third of that of the spectral multigrid method for the same number of iterations and on the same grid. To obtain the same accuracy, the CPU time of the present method is about one-tenth of the Chebyshev spectral multigrid method. Because the problem is solved on a single grid, the programming for the present method is much simpler than that for the spectral multigrid method, especially for 3D problems.

From the derivation in Section 2, the new method requires that $\nabla\sigma$ exist in the whole domain. For the 1D problem, to ensure that the iteration process converges, from (49) the maximum absolute value of $d \ln \sigma/dx$ should be less than π. When these two conditions cannot be satisfied, the new method will not work. In these cases, we may divide the whole domain into a number of subdomains so that in each subdomain $\nabla\sigma$ exists and $d \ln \sigma/dx < \pi$. Then we can solve the problem by the multidomain Chebyshev spectral method [15]. In each subdomain the problem can be solved by the new iterative method.

The Chebyshev spectral multigrid method will be superior in cases of a strong discontinuous coefficient, provided the interpolation operator is constructed to maintain continuity of $\sigma \nabla u$ [9].

6.3. *Computational Operations*

Suppose that d represents the dimension of the problem. In each iteration the present algorithm requires the solution of a forward Poisson's problem. This requires $2d(N-1)^{d+1} + d(N-1)^d$ operations (one addition and one multiplication count for one operation). The evaluation of ∇u requires $d(N-1)^{d+1}$ operations. The factorization of matrix $\tilde{d}_{ij}^{(2)}$ requires $4(N-1)^3$ operations. For 1D problems the factorization may be too expensive. For 2D and 3D problems the cost of factorization process is relatively small. The total operation count for 2D and 3D problems is of $O(N^{d+1})$. In practical computation we can build a lookup table to store the factorized matrices and the eigenvalues for commonly used N's.

ACKNOWLEDGMENTS

We thank Professor D. W. Strangway, the president of the University of British Columbia, for providing financial support and many encouragements to the first author. This work was supported by a Natural Science and Engineering Research Council of Canada (NSERC) Operating Grant 5-80642 and Professor D. W. Strangway's UBC research grant.

REFERENCES

1. A. Brandt, S. R. Fulton, and G. D. Taylor, *J. Comput. Phys.* **58**, 96 (1985).
2. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).
3. M. O. Deville and E. H. Mund, *J. Comput. Phys.* **60**, 517 (1985).
4. M. O. Deville and E. H. Mund, *SIAM J. Sci. Stat. Comput.* **11**, 311 (1990).
5. D. Gottlieb, M. Y. Hussaini, and S. A. Orszag, "Theory and Applications of Spectral Methods," in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt, D. Gotlieb, and M. Y. Hussaini (SIAM-CBMS, Philadelphia, 1984).
6. G. Gottlieb and L. Lustman, *SIAM J. Numer. Anal.* **20**, 909 (1983).
7. R. E. Lynch, J. R. Rice, and D. H. Thomas, *Numer. Math.* **6**, 185 (1964).
8. R. Peyret, *Introduction to Spectral Methods*, von Karman Institute Lecture Series 1986-04, (Rhode-Saint Genese, Belgium, 1986).
9. K. Stúben, *Algebraic Multigrid (AMG): Experiences and Comparisons* (Gesellschaft Für Mathematik Und Datenverarbeitung Mbh, Bonn, 1983.
10. R. S. Varga, *Matrix iterative analysis* (Prentice–Hall, Englewood Cliffs, NJ, 1962).
11. J. H. Wilkinson, *The Algebraic Eigenvalue Problem* (Clarendon Press, Oxford, 1965).
12. T. A. Zang, Y. S. Wong, and M. Y. Hussaini, *J. Comput. Phys.* **48**, 485 (1982).
13. T. A. Zang, Y. S. Wong, and M. Y. Hussaini, *J. Comput. Phys.* **54**, 489 (1984).
14. T. A. Zang, C. L. Streett, and M. Y. Hussaini, ICASE Report No. 89-13, 1989 (unpublished).
15. S. K. Zhao, Ph.D. thesis, University of British Columbia, 1993 (unpublished).
16. D. B. Haidvogel and T. A. Zang, *J. Comput. Phys.* **30**, 167 (1979).
17. P. Haldenwang, G. Labrosse, S. Abboudi, and M. Deville, *J. Comput. Phys.* **55**, 115 (1984).