JOURNAL OF COMPUTATIONAL PHYSICS 33, 236–248 (1979)

Direct Solution of the Biharmonic Equation Using Noncoupled Approach

MURLI M. GUPTA

Mathematics Department, George Washington University, Washington, D.C. 20052

AND

RAM P. MANOHAR

Mathematics Department, University of Saskatchewan, Saskatoon, Canada S7N OWO

Received September 6, 1978; revised January 8, 1979

The Dirichlet problem for the biharmonic equation is solved using the thirteen-point formula. The prescribed normal derivative on the boundary is replaced by two classes of boundary approximations in order to define the solution at certain fictitious node points. A direct method is used to solve the resulting system of algebraic equations. It is found that the accuracy of the numerical solution strongly depends upon the boundary approximation used, as in the coupled-equation approach. However, the cost of obtaining the solution is almost independent of the boundary approximation, unlike the coupled-equation approach.

1. INTRODUCTION

Consider the Dirichlet problem for the biharmonic equation

$$\Delta\Delta u(P) = F(P) \qquad P \in D \tag{1.1}$$

$$u(P) = f(P), \qquad \frac{\partial u}{\partial n}(P) = g(P), \qquad P \in \dot{D}, \tag{1.2}$$

where D is a closed convex domain in two dimensions and $\dot{D}$ is its boundary; $\partial u/\partial n$ represents the outward normal on $\dot{D}$.

The biharmonic equation (1.1) is frequently split into two Poisson equations

$$\Delta u(P) = v(P), \qquad \Delta v(P) = F(P), \qquad P \in D. \tag{1.3}$$

The boundary values for u and v may be written as follows:

$$u(P) = f(P), \qquad P \in \dot{D} \tag{1.4a}$$

$$v(P) = \Delta u(P) - c(u_n(P) - g(P)), \qquad P \in \dot{D}. \tag{1.4b}$$

0021-9991/79/110236-13$02.00/0

The coupled boundary-value problems (1.3)–(1.4) are equivalent to the problem (1.1)–(1.2) and have a unique solution for any nonzero value of c [10, 14].

In order to solve the biharmonic equation numerically, we introduce a uniform mesh of width h. Let D_h be the set of all mesh points inside D, $\dot{D}_h$ be the set of boundary mesh points, and $\bar{D}_h = D_h \cup \dot{D}_h$. Let u_h, v_h represent the finite-difference approximations of u, v, respectively. The Poisson equations in (1.3) are discretized using the five-point formula:

$$\begin{aligned}\Delta_h u_h = h^{-2}[&u_h(x-h, y) + u_h(x+h, y) + u_h(x, y-h)\\ &+ u_h(x, y+h) - 4u_h(x, y)]\end{aligned}$$

at each mesh point P in D_h. The boundary values of v are defined in terms of Δu which is undefined on $\dot{D}$. However, these values can be approximated using boundary approximations $\bar{\Delta}_h u_h$ defined in [5, 10]:

Let $r, \bar{r}, \underline{r} \in \dot{D}_h$ such that $\overline{r\bar{r}} = \overline{r\underline{r}} = h > 0$. Let r_p, $r_q \in D_h$ such that r, r_p, r_q lie on a straight line perpendicular to the boundary $\dot{D}$ at r and $\overline{rr}_p = ph$, $\overline{rr}_q = qh$.

A boundary approximation of order h is given by

$$\begin{aligned}v_h(r) = \bar{\Delta}_h u_h(r) = h^{-2}[&2p^{-2}u_h(r_p) + 2p^{-1}hu_n(r) + f(\bar{r})\\ &+ f(\underline{r}) - 2(1 + p^{-2})f(r)], \qquad p \geqslant 1. \qquad (1.5)\end{aligned}$$

This approximation with $p = 1$ corresponds to the conventional method of boundary approximation [4, 8, 14–18]. This approximation with $p = 2$ has also been used for computations of viscous flow problems [7, 11].

A boundary approximation of order h^2 is given by

$$\begin{aligned}v_h(r) = \bar{\Delta}_h u_h(r) = h^{-2}\Big[&2p^3\alpha u_h(r_q) - 2q^3\alpha u_h(r_p) + f(\bar{r})\\ &+ f(\underline{r}) + \frac{(p+q)}{pq}hu_n(r) - 2(1 + \alpha p^3 - \alpha q^3)f(r)\Big], \qquad (1.6)\end{aligned}$$

where

$$\alpha = p^{-2}q^{-2}(p-q)^{-1}, \qquad p \neq q; \qquad p, q \geqslant 1.$$

This approximation with $p = 2$, $q = 1$ has been briefly studied by Ehrlich [4]. Some other authors have commented on the suitability of this type of formulas [15, 18].

When $\bar{\Delta}_h$ is defined by the first-order boundary approximation (1.5), the discretization error ($e_h = u - u_h$) in l_2-norm is given by [10]

$$\| e \|_2^2 \leqslant C\left[\frac{p^2h^3}{9} M_3^2 + h^4\left(\frac{p^2(p^2+1)}{18} M_3M_4 + \frac{4}{9} M_6^2\right) + O(h^5)\right], \qquad (1.7)$$

where M_j is the maximum value of the j-th derivatives of u on the domain D. If h is sufficiently small and M_j are reasonably bounded, then

$$\operatorname*{Max}_{P\in\bar{D}_h} |e_h(P)| \leqslant O(h^{3/2}), \qquad h \to 0. \tag{1.8}$$

When Δ_h is defined by the second-order approximation (1.6), the discretization error is given by

$$\| e_h \|_2^2 \leqslant C\left[\frac{4}{9} h^4 M_6{}^2 + \frac{(pq+1)^2}{144} h^5 M_4{}^2 + O(h^6)\right]; \tag{1.9}$$

and

$$\max_{P\in\bar{D}_h} | e_h | \leqslant O(h^2), \qquad h \to 0. \tag{1.10}$$

It is clear from the above estimates that the error bounds generally increase when larger values of p and q are chosen to define the boundary values of v_h. However, when the derivatives M_j are very large and the mesh size h not too small, this trend may not hold. We illustrate this point in Section 3.

It has been shown [5] that the rate of convergence of the iterative procedure to solve the discrete form of the coupled equations (1.3)–(1.4) increases when boundary approximations with larger p and q are used. There is a tradeoff between the speed of convergence and accuracy and suitable values of p and q exist for which the convergence is faster than the conventional method (approximation (1.5) with $p = 1$) as well as the accuracy being improved. In fact, it has been recommended to choose p, q as large as desired such that $pq = O(h^{-1})$ in order to improve accuracy as well as the rate of convergence [5].

In the present paper we examine the numerical procedures for solving the biharmonic equation (1.1) without splitting it into two Poisson equations. This procedure avoids the need to define v $(= \Delta u)$ on the boundaries as well as the use of iterative procedure to solve the coupled Poisson equations (1.3)–(1.4). We discretize the equation (1.1) using the thirteen-point formula with truncation error of order h^2. This formula requires the values of u_h at certain fictitious node points outside $\bar{D}_h$. We propose boundary approximations similar to (1.5), (1.6) to define these unknown u_h values. We use a direct method to solve the resulting system of algebraic equations. The computational effort does not depend as strongly on the boundary approximation used as in the case of the coupled-equation approach [5]. However, the accuracy of the approximate solution strongly depends upon the choice of the boundary approximation. Based on the theoretical and numerical evidence, we conclude that the boundary approximations of second order with moderate values of p and q indeed provide a most accurate solution in the class of $O(h^2)$ schemes.

The thirteen-point formula has previously been studied by various authors [3, 6, 12] who devised iterative procedures to solve the discrete equations. Due to the ill-conditioned nature of the coefficient matrix, these attempts were not very successful, Some direct methods have also been proposed to solve this discrete system [1, 2, 4, 13, 16]. All of these studies utilize the reflection principle to define the unknown values of u_h outside $\bar{D}_h$. When any of the other boundary approximations are used, the coefficient matrices lose their regular structure and the above direct methods are not applicable. We use the boundary approximations from both classes of formulas and study their effect on the overall accuracy. In the last sections we consider possible improvements as well as the limitations of our method.

2. The Noncoupled Approach

The thirteen-point approximation-of the biharmonic equation (1.1) is given by

$$\begin{aligned} L_h u_{ij} = h^{-4}[&u_{ij-2} + u_{ij+2} + u_{i-2,j} + u_{i+2,j} - 8(u_{ij-1} + u_{ij+1} \\ &+ u_{i-1,j} + u_{i+1,j}) + 2(u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} \\ &+ u_{i+1,j+1}) + 20u_{ij}]. \end{aligned} \tag{2.1}$$

At each mesh point (x_i, y_j) of D_h, we write the finite-difference approximation of the biharmonic equation (1.1) as follows:

$$L_h u_h(x_i, y_j) = F(x_i, y_j). \tag{2.2}$$

When the mesh point (x_i, y_j) is adjacent to the boundary $\dot{D}_h$, then Eq. (2.2) involves at least one value of u_h which lies outside $\bar{D}_h$ (for example, if $(x_i, y_j) \in D_h$ and $(x_{i-1}, y_j) \in \dot{D}_h$, then $u_{i-2,j}$ is undefined because $(x_{i-2,}y_j) \notin \bar{D}_h$). Such undefined values of u_h are conventionally calculated by the following reflection formula [1–4, 6, 12, 13, 16]:

$$\left(\frac{\partial u}{\partial x}\right)_{i-1,j} = \frac{u_{i,j} - u_{i-2,j}}{2h} + O(h^2), \tag{2.3}$$

and the unknown value of $u_{i-2,j}$ is obtained from

$$u_{i-2,j} \approx u_{i,j} - 2h\left(\frac{\partial u}{\partial x}\right)_{i-1,j}. \tag{2.4}$$

In this paper, we consider approximations for $\partial u/\partial n$ that involve values of u_h far inside the region D_h. The first-order boundary approximation is given by

$$\frac{\partial u}{\partial n}(r) = \frac{-p}{2h}[2p^{-2}u(r_p) - u(r_1) - u(r_{-1}) + (2 - 2p^{-2})\,u(r)] + O(h^2), \qquad p \geqslant 1, \tag{2.5}$$

where $r \in \dot{D}_h$; r_1, $r_p \in D_h$ and $r_{-1} \notin \bar{D}_h$ such that $\overline{rr}_p = ph$, $\overline{rr}_1 = \overline{rr}_{-1} = h$ and

r, r_{-1}, r_1, r_p all lie on the normal at r. Using (2.5) we can define the value of $u_h(r_{-1})$ which is substituted into the left-hand side of (2.2). For example,

$$u_{i-2,j} \approx 2p^{-2}u_{i+p-1,j} - u_{ij} + (2 - 2p^{-2})\, u_{i-1,j} - 2hp^{-1}\left(\frac{\partial u}{\partial x}\right)_{i-1,j}. \tag{2.6}$$

The second-order boundary approximation of $\partial u/\partial n$ is defined by

$$\frac{\partial u}{\partial n}(r) = \frac{-pq}{2(p+q)h}\,[-2q^3\alpha u(r_p) + 2p^3\alpha u(r_q) - u(r_1) - u(r_{-1}) \\ + (2 + 2q^3\alpha - 2p^3\alpha)\, u(r)] + O(h^3); \qquad p, q \geqslant 1, \quad p \neq q \tag{2.7}$$

where $\alpha = p^{-2}q^{-2}(p-q)^{-1}$ and $\overline{rr}_q = qh$. As in the previous case, the value of $u_h(r_{-1})$ can be defined from (2.7). For example

$$u_{i-2,j} = -2q^3\alpha u_{i+p-1,j} + 2p^3\alpha u_{i+q-1,j} - u_{ij} + (2 + 2q^3\alpha - 2p^3\alpha)\, u_{i-1,j} \\ - 2\,\frac{(p+q)}{pq}\, h\left(\frac{\partial u}{\partial x}\right)_{i-1,j}. \tag{2.8}$$

We denote the first-order boundary approximation (2.5) as the $(p, 0)$ formula or the one-point formula. The second-order boundary approximation (2.7) is called the (p, q) formula or the two-point formula.

In particular, the (1, 0) formula (Eq. (2.6) with $p = 1$) is the conventional reflection formula (2.4). The (2, 0) formula is

$$u_{i-2,j} = \tfrac{1}{2}u_{i+1,j} - u_{ij} + \tfrac{3}{2}u_{i-1,j} - h(u_x)_{i-1,j}. \tag{2.9}$$

The higher-order (2, 1) formula (Eq. (2.8) with $p = 2$, $q = 1$) is given by

$$u_{i-2,j} = -\tfrac{1}{2}u_{i+1,j} + 3u_{i,j} - \tfrac{3}{2}u_{i-1,j} - 3h(u_x)_{i-1,j}. \tag{2.10}$$

The (3, 2) formula is given by

$$u_{i-2,j} = -\tfrac{4}{9}u_{i+2,j} + \tfrac{3}{2}u_{i+1,j} - u_{i,j} + \tfrac{17}{18}u_{i-1,j} - \tfrac{5}{3}h(u_x)_{i-1,j}. \tag{2.11}$$

The (3, 1) formula is given by

$$u_{i-2,j} = -\tfrac{1}{9}u_{i+2,j} + 2u_{ij} - \tfrac{8}{9}u_{i-1,j} - \tfrac{8}{3}h(u_x)_{i-1,j}. \tag{2.12}$$

The first author has shown [10] the equivalence of (2.1), (2.5) and (2.1), (2.7) to the corresponding approximations of the coupled Poisson equations (1.3) with (1.5) and (1.6), respectively. Consequently, the error bounds (1.7) and (1.9) hold for the direct discretization of the biharmonic equation with boundary approximations (2.5) and (2.7), respectively.

3. Numerical Examples

We considered several examples to study the performance of the one-point and two-point formulas for various values of p and q. Some of the examples have been studied by other authors and provide a useful basis for comparison. In each case we took the unit square $0 \leqslant x, y \leqslant 1$ as the region of integration and covered it by a mesh of uniform width h: $\{x_i = ih, y_j = jh, 1 \leqslant i, j \leqslant n; (n+1)h = 1\}$.

The thirteen-point formula (2.1) was written at each mesh point (ih, jh). When this formula needed a value of u_h on $\dot{D}_h$, the given data was substituted and the corresponding term removed to the right-hand side. When the formula needed a value of u_h outside $\bar{D}_h$, this value was defined using (2.5) or (2.7). This resulted in the modification of some of the elements of the coefficients matrix A of the system:

$$A\mathbf{u} = \mathbf{b}. \tag{3.1}$$

The matrix A is of order n^2 and has at most 13 nonzero elements in each row. The coefficient matrix varies with the boundary approximation. If $1 \leqslant p, q \leqslant 3$, then A has 13 nonzero diagonals in a band form with total bandwidth $(4n+1)$. When p or q are taken to be larger than 3, then the total bandwidth increases to $2(\gamma - 1)n + 1$, where $\gamma = \max(p, q)$. This increase in bandwidth in turn affects the core requirements as well as computing times. Since the accuracy generally deteriorates with larger values of p and q [5, 10], there is nothing to be gained by taking $p, q > 3$.

We used certain direct solvers (viz., LEQT1B, LEQT2B of IMSL and GELB of SSP) to solve the system of linear equations (3.1). These band solvers require the storage of the full band (i.e., $n^2 \times (4n+1)$ elements). As an example, with mesh size $h = 0.05$, the matrix A is of order 361×361 and the band solver requires the storage of 361×77 $(= 27797)$ elements. Unfortunately, we have been unable to obtain a sparse matrix solver which may cut down the storage requirement to $n^2 \times 13$ elements (for $h = 0.05$, 4693 elements).

Since the matrix A varies with the boundary approximations, it is not possible to

TABLE I

Storage requirements and execution times (IBM370/158).

Mesh size	Number of Unknowns	Band width	Total core requirements using LEQT2B	Execution times (sec)
1/5	16	17	48K	0.06
1/10	81	37	76K	1.25
1/16	225	61	188K	7.9
1/20	361	77	328K	20.2
1/25	576	97	612K	—
1/50	2401	197	4808K	—

preprocess the matrix decomposition for all problems. However, A has a regular band structure and it would be advantageous to develop a direct solver which will take into account this structure of the matrix.

We were able to obtain fairly accurate solutions of the matrix equation (3.1) using the band solvers LEQT1B and LEQT2B of the IMSL. We were unable to use finer mesh sizes because the storage requirements for the whole band became enormous. Typical storage requirements and execution times are given in Table I.

We now give detailed results for a selection of examples.

EXAMPLE 1.

$$\Delta\Delta u = 0. \tag{3.2}$$

TABLE II

		Maximum Error, $u - u_h$		
h	Boundary Approx. (p, q)	Direct method non-coupled equation	Iterative method coupled approach	No. of iterations coupled approach
1/5	1, 0	0.5123 (−2)[a]		
	2, 0	0.1451 (−1)		
	3, 0	0.2360 (−1)		
	2, 1	0.6990 (−3)		
	3, 1	0.1012 (−2)		
	3, 2	0.2601 (−2)		
1/10	1, 0	0.1481 (−2)		
	2, 0	0.5218 (−2)		
	3, 0	0.1006 (−1)		
	2, 1	0.8774 (−4)		
	3, 1	0.1373 (−3)		
	3, 2	0.4015 (−3)		
1/16	1, 0	0.5856 (−3)	0.59 (−3)	32
	2, 0	0.2238 (−2)	0.23 (−2)	22
	3, 0	0.4695 (−2)	0.47 (−2)	17
	2, 1	0.1240 (−4)	0.23 (−4)	42
	3, 1	0.2670 (−4)	—	—
	3, 2	0.9441 (−4)	0.10 (−3)	30
1/20	1, 0	0.3710 (−3)	0.3922 (−3)	31
	2, 0	0.1454 (−2)	0.1481 (−2)	29
	3, 0	0.3135 (−2)	0.3159 (−2)	17
	2, 1	0.4768 (−5)	0.2909 (−4)	48
	3, 1	0.9538 (−5)	0.3225 (−4)	36
	3, 2	0.4387 (−4)	0.6157 (−4)	27

[a] 0.5123 (−2) = 0.5123 × 10^{-2}.

Note: Data for $h = 1/16$ (coupled approach) is taken from [5].

The exact solution is $u = x^2 + y^2 - xe^x \cos y$. This example has been considered by Ehrlich and Gupta [5] through the coupled-equation approach. The maximum difference between the approximate and exact solutions are given in Table II.

In Table II we also give some of the results from the coupled-equation approach and corresponding number of iterations required to achieve convergence. It is noted that, for example, for $h = 1/20$ the iterative procedure for the coupled equation approach takes approximately 1.5 sec per iteration (using our package [19] on IBM 370/158) which addes up to 72 sec for the boundary approximation (2, 1). As a comparison, the direct method for the noncoupled approach takes 20.2 sec for solving the same problem (see Table I). The direct method yields a slightly more accurate solution because the iterative method depends upon the stopping criteria, whereas the direct method does not.

EXAMPLE 2.

$$\Delta\Delta u = 0. \tag{3.3}$$

The exact solution is $u = x^3 - 3y^2 + 2xy$. This example was considered by Greenspan and Schultz [8] who used the coupled-equation approach with the (1, 0) boundary approximation and $h = 0.05$. Their numerical solution agreed with the exact solution to three decimal places. Convergence was achieved after 93 iterations in 6 min of computer time on UNIVAC 1108. On our computer (IBM 370/158), 93 iterations would take approximately 140 sec of execution time compared to 20.2 sec for the direct approach. The maximum errors with various boundary approximations are given in Table III.

TABLE III

Example 2

h	p, q	Maximum error	Error in [8]
1/20	1, 0	0.2168 (−3)	~0.001
	2, 0	0.8495 (−3)	
	3, 0	0.1836 (−2)	
	2, 1	0.2861 (−5)	
	3, 1	0.2861 (−5)	
	3, 2	0.2861 (−5)	

It is noted that all two-point boundary approximations in Table III yield the same error 2.861×10^{-6}. This is expected as the two-point formulas are of order h^2 and the difference equations are exactly satisfied by the exact solution (3.3). Theoretically, the discretization error is zero in this case.

Example 3.

$$\Delta\Delta u = 8[3y^2(1-y)^2 + 3x^2(1-x)^2 + (6x^2-6x+1)(6y^2-6y+1)] \tag{3.4}$$

with the exact solution $u = x^2(1-x)^2 y^2(1-y)^2$. This example was considered by Bauer and Reiss [1] who defined a fast direct solver for the thirteen-point difference approximation (2.1) of the biharmonic equation. Bauer and Reiss used the (1, 0) boundary approximation which provides a block symmetric coefficient matrix $[A_i, B_i, C_i, D_i, E_i]$ with $A_i = E_i = I$,

$$B_i = D_i = [0, 2, -8, 2, 0], \qquad C_i = [1, -8, c_k, -8, 1].$$

The block structure is not so regular when any other of the (p, o) or the (p, q) formulas are used. Hence, the block elimination procedure of [1] is not applicable in general case.

Bauer and Reiss solved Example 3 with mesh width $h = 1/26$ and the maximum relative error was 1.13% (max. error = 0.00004414) at $x = y = 0.5$. Our results with coarser mesh ($h = 1/20$) are summarized in Table IV. The two-point formuls clearly yield better results for the same cost.

In all of the above examples, we note that the two-point formulas yield a more accurate solution than the one-point formulas. Moreover, the errors become worse when p, q increase. In each of the above cases, the maximum values M_j of the jth derivatives are reasonably bounded. We now consider an example where M_j are very large and the pattern noted above does not hold.

TABLE IV

Example 3

h	p, q	Max. error	Max. relative error
1/20	1, 0	0.7463 (−4)	1.91 %
	2, 0	0.2377 (−3)	6.09 %
	3, 0	0.4856 (−3)	12.43 %
	2, 1	0.1846 (−4)	0.47 %
	3, 1	0.1994 (−4)	0.51 %
	3, 2	0.2791 (−4)	0.71 %

Example 4.

$$\Delta\Delta u = (2\pi)^4[4\cos(2\pi x)\cos(2\pi y) - \cos(2\pi x) - \cos(2\pi y)] \tag{3.5}$$

with the exact solution $u = (1 - \cos 2\pi x)(1 - \cos 2\pi y)$.

This example was considered by Bauer and Reiss [1] who used their direct solver with the (1, 0) boundary approximation and $h = 1/26$. The maximum relative error was found to be 0.98 % at $x = y = 0.5$ (maximum absolute error = 0.03920). The results of our calculations with $h = 1/20$ are summarized in Table V.

TABLE V

Example 4

h	p, q	Max. error	Max. relative error
1/20	1, 0	0.066639	1.66 %
	2, 0	0.083653	2.09 %
	3, 0	0.132081	3.30 %
	2, 1	0.060430	1.51 %
	3, 1	0.057601	1.44 %
	3, 2	0.042403	1.06 %
	4, 1	0.054934	1.37 %
	4, 2	0.033173	0.83 %
	4, 3	0.019814	0.49 %

In this example, we note that the second-order formulas are indeed more accurate than the first-order formulas. However, the errors tend to decrease as p, q are increased! This is an unexpected trend because the (2, 1) formula is expected to be the most accurate formula in the class of second-order boundary approximations.

The discrepancy is explained by the fact that the derivatives of the exact solution u have large values and the term involving p, q in the estimate (1.9) has a negligible effect on the overall error.

In fact, in this example, $M_j = 2(2\pi)^j$ and $M_3 = 496.10$, $M_4 = 3117.09$ and $M_6 = 123057.82$. The error estimates (1.7) and (1.9) may be rewritten, respectively, as

$$\| e_h \|_2^2 \leqslant 4(2\pi)^6 h^3 C[\tfrac{1}{9} p^2 + h\{\tfrac{1}{9} p^2(p^2 + 1)\pi + \tfrac{4}{9}(2\pi)^6\}] + O(h^5) \tag{3.6}$$

and

$$\| e_h \|_2^2 \leqslant 4(2\pi)^8 h^4 C[\tfrac{4}{9}(2\pi)^4 + \tfrac{1}{144}(1 + pq)^2 h] + O(h^6). \tag{3.7}$$

In the case of the (p, o) formula, each term of the error estimate (3.6) involves p, so we expect the errors to increase when p is increased. This is true as seen in Table V. In the case of the (p, q) formula, the values of p and q do not appear in each term of the error estimate (3.7). In fact, the first term inside the brackets on the right-hand side of (3.7) has value 692.69 which is considerably large compared to the second term, unless p, q are very large. Thus, for reasonable values of p, q we expect the discretization error to be of the same order. The fact that the errors actually decrease with large p, q (two-point formulas) may be attributed to the effect of rounding errors.

Even though the (2, 1) formula is not the most accurate formula in this example, it is still more accurate than all of the one-point formulas. We also note that several of our two-point formulas (viz., (4.2) and (4, 3) formulas) give a more accurate solution with a cruder mesh ($h = 1/20$) than that obtained by Bauer and Reiss [1] with $h = 1/26$.

4. Possible Improvements

We now consider two possible methods of improving accuracy:

4.1. *Richardson Extrapolation*

Richardson extrapolation to the limit may be used to improve accuracy. The one-point formulas have discretization errors of order $h^{3/2}$, whereas the two-point formulas have errors of order h^2. We used the Richardson extrapolation with $h = 1/10$ and $h = 1/20$ and the results for Example 4 are summarized in Table VI.

TABLE VI

Richardson extrapolation on Example 4.

p, q	Max. error	Max. relative error
1, 0	0.046568	1.16%
2, 1	0.005527	0.14%
3, 1	0.008060	0.20%
3, 2	0.021153	0.53%

We note from Table VI that the (2, 1) boundary approximation again is the most accurate formula. The effects of rounding errors seem to have been smoothed out by the extrapolation procedure.

4.2. *p-Extrapolation*

The discretization error for the one point formula has the form

$$\begin{aligned}\max_{\bar{D}_h} | e_h | &= O(h^{3/2}) \\ &= \tfrac{1}{3}CpM_3h^{3/2} + O(h^2), \qquad h \to 0. \qquad (4.1)\end{aligned}$$

Since p appears explicitly in the leading term, an extrapolation on p can be carried out in order to improve accuracy. We assume that for a fixed mesh width h and two different boundary approximations $(p_1, 0)$ and $(p_2, 0)$, the solutions obtained are $u_h^1(x, y)$ and $u_h^2(x, y)$, respectively. Then,

$$u_h^1(x, y) = u(x, y) + \tfrac{1}{3}Cp_1M_3h^{3/2} + O(h^2), \tag{4.2}$$

$$u_h^2(x, y) = u(x, y) + \tfrac{1}{3}Cp_2M_3h^{3/2} + O(h^2). \tag{4.3}$$

Multiplying Eqs. (5.2) and (5.3) by $p_2/(p_2 - p_1)$ and $p_1/(p_2 - p_1)$, respectively, and subtracting, we get

$$u_h(x, y) = \frac{p_2u_h^1 - p_1u_h^2}{p_2 - p_1} = u(x, y) + O(h^2). \tag{4.4}$$

Thus the extrapolated solution $u_h(x, y)$ is of second order. The results of the above p-extrapolation on Example 4 are presented in Table VII.

TABLE VII

p-extrapolation on Example 4.

h	p_1	p_2	Max. error	Max. relative error
1/20	1	2	0.049142	1.23%
	1	3	0.033558	0.84%
	2	3	0.031312	0.78%
	1	5	0.02678	0.70%
	2	4	0.060642	1.52%
	3	6	0.273168	6.83%

From Table VII it may be noted that p-extrapolation is successful only for small values of p_1 and p_2.

Similar extrapolation for the two-point formulas does not seem feasible as the error estimate (1.9) does not contain p, q explicitly in the principal error term.

5. Limitations of the Direct Method

The major handicap in the use of direct methods for the uncoupled biharmonic equation is in the storage requirements of the coefficient matrix. At this moment we are unable to use finer mesh sizes and many boundary approximations because the available band solver routines require the storage of all the zeros within the band. When good sparse matrix packages become available and standardized, hopefully the direct methods would be applicable to a large variety of problems. We did attempt to use a recently published sparse matrix package [9] but were not successful in obtaining solutions beyond $h = 1/25$ because of the storage requirements of this solver.

6. Conclusions

We have shown how direct solvers can be used to obtain the solutions of the biharmonic equation in the noncoupled form. When a direct solver is applicable, it is much more efficient than the large variety of iterative methods available in the literature. The direct method is also more efficient than the coupled-equation approach because one need not search for the optimum parameters.

The cost of solving the discrete biharmonic equation using a direct solver is almost independent of the boundary approximation used. In such a situation, the two-point formulas are much more accurate than the one-point formulas. In particular, the (2, 1) formula is recommended as the optimum formula in order to obtain an error of order h^2.

References

1. L. Bauer and E. L. Reiss, *Math. Comp.* **26** (1972), 311–326.
2. B. L. Buzbee and F. W. Dorr, *SIAM J. Numer. Anal.* **11** (1974), 753–763.
3. S. D. Conte and R. I. Dames, *Math. Tables Aids Comp.* **12** (1958), 198–205.
4. L. W. Ehrlich, *Comm. ACM* **16** (1973), 711–714.
5. L. W. Ehrlich and M. M. Gupta, *SIAM J. Numer. Anal.* **12** (1975), 773–790.
6. G. Fairweather, A. R. Gourlay, and A. R. Mitchell, *Numer. Math.* **10** (1967), 56–66.
7. D. Greenspan, *Comput. J.* **12** (1969), 88–94.
8. D. Greenspan and D. Schultz, *Comm. ACM* **15** (1972), 347–350.
9. K. K. Gupta and S. K. Tanji, *Internat. J. Numer. Methods Engrg.* **11** (1977), 1251–1259.
10. M. M. Gupta, *SIAM J. Numer. Anal.* **12** (1975), 364–377.
11. M. M. Gupta and R. Manohar, *Internat. J. Numer. Methods Engrg.* **4** (1972), 251–260.
12. A. Hadjidimos, *Numer. Math.* **17** (1971), 301–317.
13. D. A. H. Jacobs, *J. Computational Phys.* **13** (1973), 303–315.
14. J. W. McLaurin, *SIAM J. Numer. Anal.* **11** (1974), 14–33.
15. P. J. Roache, "Computational Fluid Dynamics," Hermosa Publishers, Albuquerque, N. M., 1972.
16. A. H. Sameh, University of Illinois, Comp. Sci. Dept., Report UIUCDCS–R–74–684, 1974.
17. J. Smith, *SIAM J. Numer. Anal.* **5** (1968), 323–339.
18. J. C. Wu, *AIAA J.* **14** (1976), 1042–1049.
19. M. M. Gupta, Abstract 73T–C44, Notices Amer. Math. Soc., 1973.