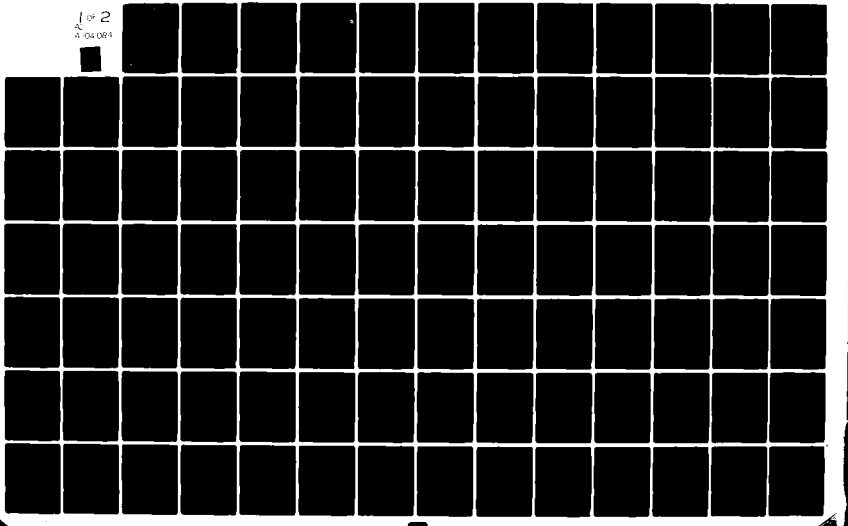
AD-A104 084 STANFORD UNIV CA DEPT OF COMPUTER SCIENCE F/G 12/1
NUMERICAL SOLUTION OF THE BIHARMONIC EQUATION.(U)
DEC 80 P E BJORSTAD N00014-75-C-1132
UNCLASSIFIED STAN-CS-80-834 NL



December 1980

LEVEL II

Report. No. STAN-CS-80-834

23

AD A104084

Numerical Solution of the Biharmonic Equation

by

Petter E. Bjørstad

Research sponsored by

Department of Energy
and
Office of Naval Research

Department of Computer Science

Stanford University
Stanford, CA 94305

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: STAN-CS-80-834	2. GOVT ACCESSION NO.: AD-A104084
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Numerical Solution of the Biharmonic Equation.	5. TYPE OF REPORT & PERIOD COVERED: Technical rept.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Petter E. Bjorstad	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-1132; DE-AT03-76ER72018
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Stanford University, Computer Science Department, Stanford, California 94304	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Mathematics Program (Code 432), Arlington, Virginia 22217	12. REPORT DATE: December 1980
	13. NUMBER OF PAGES: 140
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Unlimited	DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The numerical solution of discrete approximations to the first biharmonic boundary value problem in rectangular domains is studied. Several finite difference schemes are compared and a family of new fast algorithms for the solution of the discrete systems is developed. These methods are optimal, having a theoretical computational complexity of $O(N^2)$ arithmetic operations and requiring $N^2+O(N)$ storage locations when solving the problem on an N by N grid. Several practical computer implementations of the algorithm are

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

discussed and compared. These implementations require $aN^2 + bN^2 \log N$ arithmetic operations with $b << a$. The algorithms take full advantage of vector or parallel computers and can also be used to solve a sequence of problems efficiently. A new fast direct method for the biharmonic problem on a disk is also developed. It is shown how the new method of solution is related to the associated eigenvalue problem. The results of extensive numerical tests and comparisons are included throughout the dissertation.

It is believed that the material presented provides a good foundation for practical computer implementations and that the numerical solution of the biharmonic equation in rectangular domains from now on, will be considered no more difficult than Poisson's equation.

NUMERICAL SOLUTION
OF
THE BIHARMONIC EQUATION

by

Petter E. Bjørstad

This work was supported in part by the Norwegian Research Council for Science and the Humanities; Department of Energy Contract DE-AT03-76ER71030; and the Office of Naval Research Contract N00014-75-C-1132.

Abstract

The numerical solution of discrete approximations to the first biharmonic boundary value problem in rectangular domains is studied. Several finite difference schemes are compared and a family of new fast algorithms for the solution of the discrete systems is developed. These methods are optimal, having a theoretical computational complexity of $O(N_L^2)$ arithmetic operations and requiring $N_L^2+O(N)$ storage locations when solving the problem on an N by N grid. Several practical computer implementations of the algorithm are discussed and compared. These implementations require $aN^2 + bN^2\log N$ arithmetic operations with $b<<a$. The algorithms take full advantage of vector or parallel computers and can also be used to solve a sequence of problems efficiently. A new fast direct method for the biharmonic problem on a disk is also developed. It is shown how the new method of solution is related to the associated eigenvalue problem. The results of extensive numerical tests and comparisons are included throughout the dissertation.

It is believed that the material presented provides a good foundation for practical computer implementations and that the numerical solution of the biharmonic equation in rectangular domains from now on, will be considered no more difficult than Poisson's equation.

Acknowledgements

I want to thank my thesis advisor, Professor Gene Golub for his generous support and encouragement and for suggesting the topic of this dissertation. I thank him for many stimulating discussions introducing me to a field where he has made so many fundamental contributions. His work provided the foundation for my dissertation.

Most of all, I thank him for his friendship and hospitality making Stanford a unique place to study numerical analysis, and the years of study a rich and broad experience.

I would like to thank the other members of my committee, Professor Joseph Oliger for his encouragement, Professor Richard Cottle for his firm support when my program was in its beginning, and Professor Oscar Buneman for his enthusiasm, careful reading and help with the Cray-1 computer. Finally, I thank Professor Olof Widlund for his many comments and careful reading of the manuscript, making him an "unofficial" member of the committee.

Thanks are due to Janet Wright for running Serra House so smoothly, to Pamela Hagan for her excellent typing of this dissertation under heavy time pressure and to all other former and present members of Serra House that I learned to know. Excellent computing facilities were provided by the Stanford Linear Accellerator Center. This acknowledgement would not be complete without also extending thanks to my closest friends in these years, Sverre Frøyen, Eric Grosse and Aasmund Sudbø.

Most of all, I extend my deepest appreciation to Heidi, this work would not have been completed without her love, patience and understanding when I spent too many evenings thinking about Δ^2.

I dedicate this work to my parents Bitt and Torger for giving me continuous motivation, support and encouragement since 1950.

December 1980, Petter Bjørstad

* * *

This research was supported by the following institutions; The Norwegian Research Council for Science and the Humanities, Department of Energy Contract DE-AT03-76ER71030, and Office of Naval Research Contract N00014-75-C-1132.

Table of Contents

CHAPTER		PAGE
I	The Continuous Problem	1
II	The Discrete Approximation	9
	2.1 Finite difference schemes	9
	2.2 Discretization error estimates	16
	2.3 Numerical study of discretization errors	20
	2.4 A brief survey of other methods	31
III	An $O(N^2)$ Method for the Solution of the First Biharmonic Problem	36
IV	Computer Algorithms	61
	4.1 An algorithm using Fourier transform and penta-diagonal linear systems	61
	4.2 An algorithm based on Fourier transformations	68
	4.3 An efficient direct method	73
	4.4 The solution of several problems on the same grid using conjugate gradients	75
	4.5 Algorithms for vector and parallel computers	80
	4.6 Roundoff errors	85
	4.7 Efficient solution of the discrete system when the cubic extrapolation scheme is used near the boundary	88
	4.8 Efficient solution of the biharmonic equation in a disk	92
	4.9 Conformal mapping and the solution of the biharmonic equation on more general domains	95

Table of Contents (contd.)

CHAPTER		PAGE
V	Applications	98
	5.1 The eigenvalue problem for the biharmonic operator	98
	5.2 Navier Stokes equation	104
	APPENDIX I	109
	APPENDIX II	112
	APPENDIX III	121
	BIBLIOGRAPHY	124

CHAPTER I

THE CONTINUOUS PROBLEM

Let Ω be an open set in R^2 with boundary $\partial\Omega$. Consider the following problem:

$$\begin{aligned} \Delta^2 u(x,y) &= f(x,y) \qquad (x,y) \in \Omega \\ u(x,y) &= g(x,y) \qquad (x,y) \in \partial\Omega \\ u_n(x,y) &= h(x,y) \qquad (x,y) \in \partial\Omega \end{aligned} \tag{1.1}$$

where u_n denotes the exterior normal derivative on $\partial\Omega$.

This thesis will develop efficient numerical methods for the above problem when Ω is a rectangle or a circular disk. The algorithms are optimal, requiring $O(N^2)$ arithmetic operations and $O(N^2)$ storage locations for computing an approximate solution at N^2 discrete grid-points.

In this Chapter some physical problems that lead to equations like (1.1) will be described together with a few mathematical properties relevant for the construction of numerical methods. Discrete approximations to (1.1) are discussed in Chapter II, and the theory behind the numerical algorithm for the rectangular domain is developed in Chapter III. Chapter IV discusses the implementation of numerical algorithms and the design of computer programs. Some numerical results for a few applications of the algorithms to some difficult problems are presented in Chapter V.

Equation (1.1) is called the (first) Dirichlet boundary value problem for the biharmonic operator

$$\Delta^2 \equiv \nabla^4 = \frac{\partial^4}{\partial x^4} + 2\,\frac{\partial^4}{\partial x^2\,\partial y^2} + \frac{\partial^4}{\partial y^4} \tag{1.2}$$

and this problem arises in several fields of applied mathematics. Classical examples occur in elasticity theory and in the theory of fluid mechanics.

In linear elasticity $u(x,y)$ can represent the Airy stress function or as in the theory of thin plates, the vertical displacement due to an external force. In the latter case equation (1.1) represents a "clamped plate" where f is the external load. Another closely related case is that of a "supported plate" where the boundary conditions in (1.1) are replaced by

$$\begin{aligned} u(x,y) &= g(x,y) \qquad (x,y) \in \partial\Omega \\ \sigma\Delta u(x,y) + (1-\sigma)u_{nn}(x,y) &= h(x,y) \qquad (x,y) \in \partial\Omega \end{aligned} \tag{1.3}$$

where u_{nn} is the second normal derivative and σ is a material constant called Poisson's ratio.

When Ω is a polygon, this is equivalent to a problem (with data depending on σ) of the form:

$$\begin{aligned} -\Delta v &= f \quad \text{in } \Omega \\ v &= -h \quad \text{on } \partial\Omega \\ -\Delta u &= v \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned} \tag{1.4}$$

where $v = -\Delta u$ has been introduced. The original fourth order equation has been split into two Poisson problems. There exist many reliable computer programs that can be used to solve (1.4) in an efficient way both for special geometries (Swarztrauber and Sweet [1975]), and in more general domains (Proskurowski [1978]). It is important to notice that the only difference between (1.1) and (1.4) is that different boundary data have

been specified.

The theory of thin plates allowing large vertical displacements, leads to a coupled pair of nonlinear equations known as von Kármáns equations:

$$\begin{aligned} \Delta^2 u &= [u,v] + f && \text{in } \Omega \\ u &= g_1 && \text{on } \partial\Omega \\ u_n &= h_1 && \text{on } \partial\Omega \\ \Delta^2 v &= -[u,u] && \text{in } \Omega \\ v &= g_2 && \text{on } \partial\Omega \\ v_n &= h_2 && \text{on } \partial\Omega \end{aligned} \tag{1.5}$$

where

$$[u,v] = \frac{\partial^2 u}{\partial x^2}\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 u}{\partial y^2}\frac{\partial^2 v}{\partial x^2} - 2\frac{\partial^2 u}{\partial x\,\partial y}\frac{\partial^2 v}{\partial x\,\partial y} \quad .$$

Here u represents the vertical displacement of the plate, v is the Airy stress function and f is the external force on the plate. An efficient method for solving linear problems involving the biharmonic operator (with the appropriate boundary conditions) can be very valuable in iterative methods for solving more difficult problems of this type.

References describing equations involving the biharmonic operator in elasticity include Landau and Lifchitz [1970], Muskhelishvili [1963], Sokolnikoff [1946], Kupradze [1965] and Kalandiya [1975]. More recent texts on finite elements methods, Strang and Fix [1973], Zienkiewicz [1977] and Ciarlet [1978] provide additional information.

In fluid mechanics, equation (1.1) describes the streamfunction $u(x,y)$ of an incompressible two-dimensional creeping flow (Reynolds number zero). Efficient numerical methods for this problem can also be

used when trying to solve the nonlinear Navier Stokes equation describing incompressible flow at nonzero Reynolds number. The biharmonic operator appears linearly in this equation when using the streamfunction formulation. For more details on fluid mechanics applications see Landau and Lifchitz [1959] and Temam [1977].

The remaining part of this Chapter will summarize various mathematical results for the biharmonic operator Δ^2 and equation (1.1).

i) Variational forms.

Two distinct bilinear forms can be associated with problem (1.1) (Agmon [1965, p. 150]):

$$a_1(u,v) = \int_\Omega \Delta u \, \Delta v \, dx \, dy$$

$$a_2(u,v) = \int_\Omega \left[\left(\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2}\right)\left(\frac{\partial^2 v}{\partial x^2} - \frac{\partial^2 v}{\partial y^2}\right) + 4 \frac{\partial^2 u}{\partial x \, \partial y} \frac{\partial^2 v}{\partial x \, \partial y} \right] dx \, dy \quad .$$

The weak form of (1.1) corresponding to the clamped plate problem in elasticity is

$$a_3(u,v) = f(v) \qquad \forall\, v \in H_0^2(\Omega)$$

where

$$a_3(u,v) = \frac{1+\sigma}{2} a_1(u,v) + \frac{1-\sigma}{2} a_2(u,v)$$

representing the strain energy of the plate, and

$$f(v) = \int_\Omega f \, v \, dx \, dy \quad .$$

For additional material see Ciarlet [1977].

ii) Existence and Uniqueness.

If $f \in L^2(\Omega)$, $g \in H^{3/2}(\partial\Omega)$, $h \in H^{1/2}(\partial\Omega)$ and $\partial\Omega$ is sufficiently smooth, then there exists a unique (weak) solution $u \in H^2(\Omega)$ of

problem (1.1). (Lions and Magenes [1972].)

iii) Simplification of equation (1.1).

Assuming that (1.4) can be solved, there is no loss of generality to take $f = g = 0$ when discussing equation (1.1). This follows by letting $u = u_1 + u_2$ where u_1 solves (1.4). The equation for u_2 is then of the desired form.

iv) The biharmonic operator under a conformal coordinate transformation.

Assume that $s : C \to C$ maps a region Ω_z in the z-plane conformally onto a region Ω_w in the w-plane, and let Δ_z and Δ_w denote the Laplace operators in the two regions. Then

$$\Delta_z^2 u(z) = |s'(z)|^2 \Delta_w(|s'(s^{-1}(w))|^2 \Delta_w u(s^{-1}(w))) \quad . \qquad (1.6)$$

This transformation is useful when the map s from a simple (computational) domain Ω_z to the (physical) domain Ω_w is known or can be computed. References using conformal mapping and complex variable techniques include Muskhelishvili [1953] and Kantorovich and Krylov [1958].

v) Biharmonic functions.

Any function $u(x,y)$ satisfying equation (1.1) with $f \equiv 0$ is called biharmonic. Any biharmonic function u can be written

$$u(z) = \mathrm{Re}[\bar{z}\, \phi(z) + \chi(z)]$$

where ϕ and χ are analytic functions. Conversely, given ϕ and χ analytic, the above expression defines a biharmonic function u. This representation is due to Goursat [1898]. If Ω is starshaped and u is biharmonic, then

$$u = r^2 v + w \qquad (1.7)$$

where v and w are harmonic functions and $r^2 = x^2 + y^2$. (Tychonoff and Samarski [1959, p. 388], see also Kalandiya [1975].)

vi) Explicit solution of (1.1) in a disk.

Assume that $f = 0$ in (1.1) and that Ω is a disk of radius R. Then

$$u(r,\theta) = \frac{1}{2\pi R}(R^2-r^2)^2\left[\int_0^{2\pi} \frac{g(R,\alpha)(R-r\cos(\alpha-\theta))}{(R^2+r^2-2Rr\cos(\alpha-\theta))^2}\,d\alpha - \frac{1}{2}\int_0^{2\pi} \frac{h(R,\alpha)}{R^2+r^2-2Rr\cos(\alpha-\theta)}\,d\alpha\right] \tag{1.8}$$

(Tychonoff and Samarski [1959, p. 389].)

vii) Majorization of biharmonic functions in terms of the boundary data.

Despite the close connection between harmonic and biharmonic functions there is no maximum principle for biharmonic functions. The following result is due to Miranda [1948].

Assume $f = g = 0$ in (1.1), if the boundary $\partial\Omega$ is sufficiently smooth and u has continuous first partials in Ω, then

$$|u(x,y)| \leq (2\phi(x,y))^{\frac{1}{2}} \max_{\partial\Omega} |h(x,y)|$$

where $\Delta\phi = -1$ in Ω, $\phi = 0$ on $\partial\Omega$.

Extensions of this result to the case where g is nonzero are given by Rosser [1980] for circular and rectangular domains.

Another type of a priori inequality is given by

$$\int_\Omega u^2\,dxdy \le \alpha_1 \int_\Omega (\Delta^2 u)^2 dxdy + \alpha_2 \int_{\partial\Omega} u^2 ds + \alpha_3 \int_{\partial\Omega} u_n^2 ds + \alpha_4 \int_{\partial\Omega} u_t^2 ds\ .$$

Here u_t is the tangential derivative and α_1, α_2, α_3 and α_4 are (in principle) computable constants depending on the domain. This inequality holds for any sufficiently smooth function. (Sigillioto [1976]).

viii) The coupled equation approach.

Consider the following algorithm for solving (1.1). Let $\lambda^0 = 0$, then for $n = 0,1,2,..$

$$\begin{aligned} -\Delta v^n &= f && \text{in } \Omega \\ v^n &= \lambda^n && \text{on } \partial\Omega \\ -\Delta u^n &= v^n && \text{in } \Omega \\ u^n &= g && \text{on } \partial\Omega \end{aligned}$$

$$\lambda^{n+1} = \lambda^n + \rho\left(\frac{\partial u^n}{\partial n} - h\right) \qquad 0 < \rho < 2/\mu$$

where

$$\mu = \max_{\substack{v \in H^2(\Omega) \cap H_0^1(\Omega) \\ v \neq 0}} \int_{\partial\Omega} |v_n|^2\,ds \Big/ \iint_\Omega |\Delta v|^2\,dxdy\ .$$

For sufficiently smooth data it can be proved that

$$\lim_{n \to \infty} \{u^n, v^n\} = \{u, -\Delta u\}\ ,$$

see MacLaurin [1974] or Glowinski-Lions-Tremolieres [1976, Chap. 4].

ix) Relation between u_n and Δu on $\partial\Omega$.

Let $v \equiv -\Delta u$ and assume that $\lambda \equiv v|_{\partial\Omega}$ is known. Equation (1.1)

can then be solved as two decoupled problems. Let A denote the linear operator mapping λ to u_n. Glowinski and Pironneau [1979] proved that A is a symmetric, strongly elliptic operator mapping $H^{-\frac{1}{2}}(\partial\Omega)$ to $H^{\frac{1}{2}}(\partial\Omega)$. This is the basis for the mixed finite element method they propose for problem (1.1).

CHAPTER II

THE DISCRETE APPROXIMATION

This Chapter will discuss discrete approximations to the continuous biharmonic problem. The Chapter consists of four sections. First, a few possible finite difference schemes will be introduced. Necessary modifications at gridpoints near the boundary are discussed and some properties of the resulting linear systems of equations are mentioned.

The second part summarizes known theoretical results on the convergence of the finite difference solution to that of the continuous problem. Several conflicting results can be found in the literature and the review of this material is intended to clarify the knowledge of this subject.

The algorithms proposed in this thesis make it feasible to solve discrete approximations on much finer grids than previous methods could handle using limited computer resources. This made it possible to perform fairly extensive tests, solving a class of test problems over a wide range of grids in order to numerically test the theoretical convergence rates and compare some of the proposed approximations. The third section of the Chapter contains a summary of the calculations performed and some of the resulting conclusions.

The last section contains a short discussion of previously proposed methods for solving discrete approximations to the biharmonic equation.

2.1 Finite difference schemes.

Most finite difference schemes proposed for the biharmonic equation have only been applied to regions made up of unions of rectangles. For more general regions Bramble [1966] proposed an elegant scheme which

employs the 13-point stencil. However, due to the difficulties in handling the difference approximation close to a curved boundary, more general domains should usually be treated within the framework of the finite element method. The study of finite difference schemes and the efficient numerical solution of the resulting equations derived from regular geometries is still useful for at least two reasons. There are several important problems where the geometry is regular or where it is convenient to make a coordinate transformation from the physical region to a computational domain with a simple geometry. In addition, efficient numerical methods for regular grids can contribute to the development of fast methods for solving finite element equations resulting from triangulations that are regular in the interior of a more general region. This line of development is already very evident in the work of Proskurowski and Widlund [1976], [1980] on second order elliptic equations.

The following discussion will be restricted to a rectangular region R. Let R be covered by a uniform grid such that the boundary of R falls on gridlines. This is illustrated in Figure 2.1, which also defines three disjoint sets of gridpoints, R_h, R_h^* and $\mathring{R}_h$:

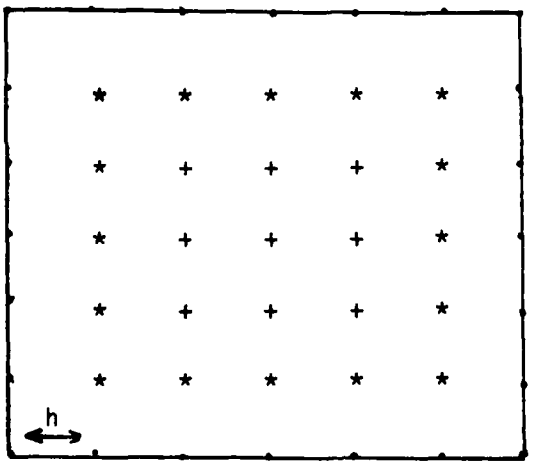


Figure 2.1. The uniform discretization of R.

$R_h = \{+\}$, the set of all gridpoints having only interior points as neighbors.

$R_h^* = \{*\}$, the set of all interior gridpoints having at least one neighbor on the boundary.

$\dot{R}_h = \{\cdot\}$ the set of boundary gridpoints.

The finite difference approximations are most conveniently described using stencils. For example,

$$\Delta_5 u \equiv \frac{1}{h^2} \begin{array}{|ccc|} \hline & 1 & \\ 1 & -4 & 1 \\ & 1 & \\ \hline \end{array}\, u = \Delta u + \frac{h^2}{12}(D_1^4 + D_2^4)u + O(h^4) \qquad (2.1)$$

defines the usual 5-point approximation to the Laplace operator and shows that this approximation has a local truncation error of order h^2. ($D_i \equiv \frac{\partial}{\partial x_i}$, i = 1 or 2). The classical 13-point approximation for the biharmonic operator is most easily derived by applying the above 5-point operator twice:

$$\Delta_{13}^2 u \equiv \Delta_5(\Delta_5 u) = \frac{1}{h^4} \begin{array}{|ccccc|} \hline & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \\ \hline \end{array}\, u =$$

$$\Delta^2 u + \frac{h^2}{6}(D_1^6 + D_1^4 D_2^2 + D_1^2 D_2^4 + D_2^6)u + O(h^4) \quad . \qquad (2.2)$$

(The operator $\Delta_5(\Delta_5 u)$ is formed by first forming $\Delta_5(\Delta u)$ and then substituting for $\Delta^2 u$ using 2.1). Unlike the discrete Laplace operator which can be applied to all interior gridpoints $P \in R_h \cup R_h^*$, this

operator is only well defined on the points $P \in R_h$. Two alternative approximations of $\Delta^2 u(P)$ will be considered for $P \in R_h^*$.

i) Quadratic extrapolation.

Use the normal derivative boundary condition at the point $Q \in R_h$ nearest $P \in R_h^*$ to formally get a local $O(h^3)$ accurate extrapolated value at the "missing" (exterior) point in the stencil. This results in a stencil of the form

$$\Delta_q^2 u(P) \equiv \frac{1}{h^4} \left[\begin{array}{ccccc} & & 1 & & \\ & 2 & -8 & 2 & \\ -8 & 21 & -8 & 1 & \\ & 2 & -8 & 2 & \\ & & 1 & & \end{array} \quad u(P) + 2hu_n(Q) \right] = \Delta^2 u(P) + O(h^{-1}) \tag{2.3}$$

when applied to a point $P \in R_h^*$ near the left boundary. Notice that u_n always denotes the exterior normal derivative evaluated at the boundary. A similar procedure (eliminating two exterior points) when $P \in R_h^*$ is a cornerpoint results in a weight of 22 at the center point of the stencil.

ii) Cubic extrapolation.

Using the same approach as in i), but performing a cubic line-extrapolation results in an $O(h^4)$ accurate approximation of the "missing" point. The discrete biharmonic operator becomes

$$\Delta_c^2 u(P) \equiv \frac{1}{h^4} \left[\begin{array}{ccccc} & & 1 & & \\ & 2 & -8 & 2 & \\ -8 & 23 & -8.5 & 1 & \\ & 2 & -8 & 2 & \\ & & 1 & & \end{array} \quad u(P) + 3hu_n(Q) - \frac{3}{2} u(Q) \right] = \Delta^2 u(P) + O(1) \tag{2.4}$$

at a point $P \in R_h^*$ near the left boundary. Notice that this leads to an unsymmetric coefficient matrix.

Gupta [1975] considered two families of boundary approximations of the above form, but depending on integer parameters indicating which interior points to use in the extrapolation. His iterative method converged faster if points further away from the boundary were chosen. This clearly results in larger truncation errors. Since the algorithms in this thesis can handle the approximations that furnish the smallest truncation errors, only these two choices will be considered. (The quadratic and cubic extrapolation near the boundary is equivalent to the schemes $p = 1$ and $p = 2$, $q = 1$ respectively in the notation of the above author.)

Glowinski [1973] made the observation that the 13-point finite difference scheme combined with quadratic extrapolation near the boundary is equivalent to solving the biharmonic equation using a mixed finite element method and piecewise linear elements in the triangulation shown in Figure 2.2.

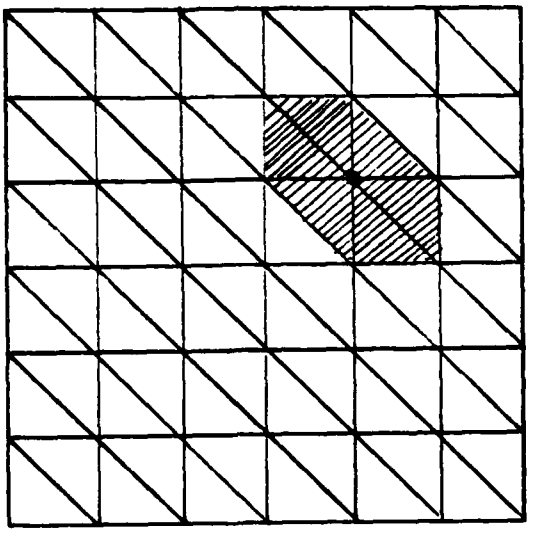


Figure 2.2. Finite element triangulation corresponding to the 13-point difference stencil.

There are many alternative finite difference approximations that can be derived for the biharmonic operator. By rotating the coordinate system $\pi/4$ the approximation

$$\Delta_x u \equiv \frac{1}{2h^2}\begin{bmatrix} 1 & & 1 \\ & -4 & \\ 1 & & 1 \end{bmatrix} u = \Delta u + \frac{h^2}{12}(D_1^4 + 6D_1^2D_2^2 + D_2^4)u + O(h^4) \qquad (2.5)$$

to the Laplace operator follows from the 5-point stencil given earlier. From this operator an alternative 13-point approximation is obtained:

$$\Delta_{xx}^2 u \equiv \Delta_x(\Delta_x u) = \frac{1}{4h^4}\begin{bmatrix} 1 & & 2 & & 1 \\ & -8 & & -8 & \\ 2 & & 20 & & 2 \\ & -8 & & -8 & \\ 1 & & 2 & & 1 \end{bmatrix} u = \Delta^2 u$$

$$+ \frac{h^2}{6}(D_1^6 + 7D_1^4 D_2^2 + 7D_1^2 D_2^4 + D_2^6)u + O(h^4) \quad . \qquad (2.6)$$

This stencil is not as convenient since it depends on twice as many points at a distance $2h$ from the center point. Combining the two approximations to the Laplacian results in a 17-point approximation.

$$\Delta_{17}^2 \equiv \Delta_x(\Delta_5 u) = \Delta_5(\Delta_x u) = \frac{1}{2h^4}\begin{bmatrix} & & 1 & & 1 & & \\ 1 & & -4 & -2 & -4 & & 1 \\ & & -2 & 16 & -2 & & \\ 1 & & -4 & -2 & -4 & & 1 \\ & & 1 & & 1 & & \end{bmatrix} u \qquad (2.7)$$

$$= \Delta^2 u + \frac{h^2}{6}(D_1^6 + 4\, D_1^4 D_2^2 + 4\, D_1^2 D_2^4 + D_2^6)u + O(h^4) \quad .$$

By taking suitable linear combinations of the above stencils for the biharmonic operator it is possible to derive approximations that can be

used to construct higher order schemes. For example

$$(\frac{1}{3}\Delta_{13}^2 + \frac{2}{3}\Delta_{17}^2)u = \Delta^2 u + \frac{h^2}{6}\Delta^3 u + O(h^4) \quad . \tag{2.8}$$

Combining this with the idea of also forming differences on the right hand side of the equation (Mehrstellenverfahren, Collatz [1955]) yields a locally fourth order accurate approximation that has been studied by Zurmühl [1957]:

$$\frac{1}{h^4}\begin{bmatrix} & 1 & 1 & 1 & \\ 1 & -2 & -10 & -2 & 1 \\ 1 & -10 & 36 & -10 & 1 \\ 1 & -2 & -10 & -2 & 1 \\ & 1 & 1 & 1 & \end{bmatrix} u = \frac{1}{2}\begin{bmatrix} & 1 & \\ 1 & 2 & 1 \\ & 1 & \end{bmatrix}\Delta^2 u + O(h^4) \quad . \tag{2.9}$$

Zurmühl derives rather complicated stencils that can be applied to points $P \in R_h^*$ having local truncation error $O(h^3)$. Based on the material in this Chapter (Section 2.2 and 2.3) it is likely that less accurate approximations near the boundary would be sufficient.

It should be noticed that all the linear systems of equations derived from the above stencils (ignoring the irregularity caused by the special boundary approximations) can be efficiently solved using for example the fast Fourier transform (Henrici [1979]).

The systems of linear equations derived from the finite difference approximations discussed so far, are all positive definite with a condition number proportional to h^{-4}. Due to the special approximations used for the points $P \in R_h^*$, the matrices are often not symmetric, but they can usually be considered as perturbed symmetric matrices.

The matrices do not possess property A (Young [1972]). Finite difference approximations of the biharmonic operator that lead to linear

systems having property A have been considered by Tee [1963] and Tang [1964] for rectangular and hexagonal meshes respectively. Proper treatment of the points near the boundary remains a serious problem with these schemes since inaccurate boundary approximations must be used in order not to destroy the matrix structure. (These matrices have interesting block properties; see Parter [1959], Chapter 14 of Young [1972], Buzbee, Golub and Howell [1977].)

2.2 Discretization error estimates.

There exist very few papers in the literature discussing the global discretization errors of the finite difference schemes presented in the previous section. This is in contrast to the case of second order elliptic problems where the theory is well understood. The main reason for this is probably the fact that there is no maximum principle for higher order equations, while the maximum principle valid both in the continuous and discrete case for second order problems provides an important tool for the analysis.

The first results proving that the 13-point approximation (2.2) converges to the solution of the continuous problem, was given by Courant, Friedrichs and Lewy [1928]. The main references for this section are Bramble [1966] and Zlámal [1967]. The more recent analysis by Gupta [1975] is based on the above paper by Zlámal, but some of the discretization error estimates given can be improved.

Let v denote a function defined on each gridpoint $P(x,y) \in R$ ($R = R_h \cup R_h^* \cup \hat{R}_h$) and extended by the value zero outside R. The following norms and notation will be used in this section:

$$v_x(x,y) \equiv \frac{1}{h}(v(x+h,y) - v(x,y))$$
$$v_{xx}(x,y) \equiv \frac{1}{h}(v_x(x,y) - v_x(x-h,y)) \tag{2.10}$$

and similarly for v_y and v_{yy}.

$$\|v\|_0^2 \equiv h^2 \sum_{P\in R} v(P)^2$$
$$\|v\|_1^2 \equiv \|v\|_0^2 + \|v_x\|_0^2 + \|v_y\|_0^2 \tag{2.11}$$
$$\|v\|_2^2 \equiv \|v\|_0^2 + \|v_x\|_1^2 + \|v_y\|_1^2 \quad .$$

The norms of the restriction of v to points $P \in R_h^*$ or $P \in R_h$ are defined by:

$$\|v\|_{0,R_h^*}^2 \equiv h \sum_{P\in R_h^*} v(P)^2$$
$$\|v\|_{0,R_h}^2 \equiv h^2 \sum_{P\in R_h} v(P)^2 \quad . \tag{2.12}$$

The norms containing the discrete derivatives are defined in a similar way as above.

The following lemma holds for any mesh function v vanishing outside R.

Lemma 2.1 (Discrete a priori inequalities.)

(i) $$\|v\|_0 \le c(\|v_x\|_0^2 + \|v_y\|_0^2)^{\frac{1}{2}}$$

(ii) $$\max_{P\in R} |v(P)| \le c|\log h^{-1}|^{\frac{1}{2}}(\|v_x\|_0^2 + \|v_y\|_0^2)^{\frac{1}{2}}$$

(iii) $$\|v\|_1 \le c(h^{-1}\|v\|_{0,R_h^*} + \|\Delta_{13}^2 v\|_{0,R_h})$$

for some constant c independent of h.

This is proved in Bramble [1966], an alternative proof of (iii) is

given by Kuttler [1971].

Lemma 2.2 (Extrapolation near the boundary).

Let v be any mesh function vanishing outside R. If the approximation defined by (2.2) and (2.3) is used, the following estimates hold:

i) $$\|v\|_{0,R_h^*} + h^{3/2}\|v\|_0 \le c\, h^{3/2}[h^{5/2}\|\Delta_q^2 v\|_{0,R_h^*} + \|\Delta_{13}^2 v\|_{0,R_h}]$$

ii) $$\|v\|_2 \le c(h^5\|\Delta_q^2 v\|^2_{0,R_h^*} + \|\Delta_{13}^2 v\|^2_{0,R_h})^{\frac{1}{2}}$$

for some constant c independent of h.

Inequality i) can be found in Kuttler [1971], ii) is proved by Zlámal [1967]. Inequality ii) also holds if Δ_q^2 is replaced by Δ_c^2 (2.4), it can therefore be used to analyze the case where cubic extrapolation is used near the boundary.

Let u_h denote the solution to a finite difference approximation of the first biharmonic boundary value problem using one of the discretizations defined in the previous section. Let u be the continuous solution of the problem and assume that $u \in C^{(6)}(R)$. Lemma 2.1 and 2.2 can now be used to estimate the global discretization error.

Theorem 2.1

Assume the 13-point operator Δ_{13}^2 is used on R and let c denote some constant independent of h. Then

i) $$\max_{P\in R}|u_h(P) - u(P)| \le c|\log h^{-1}|^{\frac{1}{2}} h^2$$

ii) $$\|u_h(P) - u(P)\|_1 \le c\, h^2$$

iii) $$\|u_h(P) - u(P)\|_2 \le c\, h^{3/2}$$

iv) $$\|u_h(P) - u(P)\|_{0,R_h^*} \leq c\, h^3$$

if the quadratic extrapolation scheme Δ_q^2 is used on R_h^*, and

v) $$\max_{P\in R} |u_h(P) - u(P)| \leq c\, h^2$$

vi) $$\max_{P\in R}(|(u_h(P) - u(P))_x| + |(u_h(P) - u(P))_y|) \leq c\, |\log h^{-1}|^{\frac{1}{2}}\, h^2$$

vii) $$\|u_h(P) - u(P)\|_2 \leq c\, h^2$$

if the cubic extrapolation scheme Δ_c^2 is used on R_h^*.

Proof:

The local truncation error of Δ_{13}^2 is $O(h^2)$, while Δ_q^2 has local truncation error $O(h^{-1})$ and Δ_c^2 is $O(1)$ (see section 2.1). Using this and i) of Lemma 2.2 gives iv). Combining this with ii) and iii) in Lemma 2.1 gives i) and ii). Statement iii) and vii) follows from ii) of Lemma 2.2. Using vii) and ii) of Lemma 2.1 gives vi). Finally, the discrete Sobolev inequality (Sobolev [1940]) applied to vii) proves v).

Remarks:

i) The above results i) - iv) still hold in a more general region with curved boundary using a suitable generalization of the quadratic extrapolation scheme. (Bramble [1966], Zlámal [1967]).

ii) It is unsatisfactory that Lemma 2.1 and 2.2 involve specific discretizations. More general proofs would make it possible to estimate the global discretization error of a given finite difference scheme from the local truncation errors. Both theoretical and computational evidence make it reasonable to believe that the lemmas hold for a wider class of approximations.

iii) Notice that the error near the boundary estimated in iv), is $O(h^3)$, an order of magnitude smaller than the overall error despite a local truncation error of $O(h^{-1})$ when using $\Delta_q^2 u(P)$ for $P \in R_h^*$. Strang and Fix [1973, p. 202] discusses a similar phenomenon for second order equations.

iv) The only important difference between quadratic and cubic boundary approximation is the estimates for the discrete second derivatives (iii) and (vii). This can be significant in several applications where Δu represents an essential physical quantity. (See also section 2.3).

v) Gupta [1975] used the discrete Sobolev inequality on iii) and obtained the weaker result $O(h^{3/2})$ instead of i).

vi) Some of the estimates in Theorem 2.1 do not seem to be sharp, see the numerical evidence in section 2.3.

vii) If $u_h^{(k)}$ is an eigenvector of the discrete biharmonic operator defined using quadratic extrapolation near the boundary, and $\lambda_h^{(k)}$ is the corresponding eigenvalue, then

$$\max_{P \in R} |u_h^{(k)}(P) - u^{(k)}| \leq c |\log h^{-1}|^{\frac{1}{2}} h^2$$

$$|\lambda_h^{(k)} - \lambda^{(k)}| \leq c\, h^2$$

provided the exact eigenfunction $u^{(k)} \in C^{(6)}$. This result is due to Kuttler [1971].

2.3 Numerical study of discretization errors.

In this section the results of numerical calculations using the finite difference methods from 2.1, are compared with the theoretical results of 2.2. The new fast computer algorithms developed in Chapter III and described in more detail in Chapter IV, make it possible to solve problems for a wide range of grids. The asymptotic behavior of the

discretization error as h tends to zero can then be investigated. The 10 testproblems listed in Appendix III were used. Scattered results for several of these problems on coarse grids can be found in the literature (see Appendix III). Each problem was solved in the unit square $0 \leq x,y \leq 1$, using a uniform grid. The results will be presented in tables like the one below.

	Solution		First derivative		Second derivative	
h_1/h_2	Max	L_2	Max	L_2	Max	L_2
	R_1	R_2	R_3	R_4	R_5	R_6

where h_1 and h_2 specifies two different grids, and

$$R_1 = \frac{\log\left(\dfrac{\max\limits_{P\in R}|u(P) - u_{h_1}(P)|}{\max\limits_{P\in R}|u(P) - u_{h_2}(P)|}\right)}{\log\left(\dfrac{h_1}{h_2}\right)} .$$

If the discretization error behaves like

$$\max_{P\in R}|u(P) - u_h(P)| \sim c_1 h^{\alpha_1} + c_2 h^{\alpha_2} + \ldots \quad (\alpha_2 > \alpha_1)$$

then R_1 will represent a computed approximation to α_1 (assuming $c_1h^{\alpha_1} >> c_2h^{\alpha_2}$). Define $e(P) \equiv u(P) - u_h(P)$ for $P \in R$. R_i, $i = 2,3,4,5,6$ is then defined in the same way as R_1 using the following norms:

$$R_2 \quad : \quad h\left[\sum_{P\in R} e(P)^2\right]^{\frac{1}{2}}$$

$$R_3 \quad : \quad \max_{P\in R}\left[|e_x(P)| \ , \ |e_y(P)|\right]$$

$$R_4 : h\left[\sum_{P\in R} (e_x(P)^2 + e_y(P)^2)\right]^{\frac{1}{2}}$$

$$R_5 : \max_{P\in R}\left[|e_{xx}(P)| , |e_{yy}(P)| , |e_{xy}(P)|\right]$$

$$R_6 : h\left[\sum_{P\in R} (e_{xx}(P)^2 + e_{yy}(P)^2 + 2\, e_{xy}(P)^2)\right]^{\frac{1}{2}} .$$

Note that the discrete derivatives of the error $e(P)$;

$$e_x(P),\ e_y(P),\ e_{xx}(P),\ e_{yy}(P) \text{ and } e_{xy}(P)$$

have been formed using centered ($O(h^2)$ accurate) differences. The same norms will be used when considering the boundary layer $P \in R_h^*$ except that the factor h is replaced by $h^{\frac{1}{2}}$ in R_2, R_4, and R_6.

Remark.

The discrete derivatives formed by centered differences of the computed pointwise error in the solution have been computed. An alternative would be to compare the finite difference approximations obtained from the computed solution with the exact derivatives. The two methods give the same information as long as $R_i \leq 2$. Since the discretization error due to the finite difference approximation of the continuous problem is best studied using the first method, only these results are presented.

First, the 13-point formula combined with quadratic extrapolation will be considered. Problem 1 is solved exactly by the method, results for problems 2, 7 and 10 are given in Figure 2.3.

	Solution		First derivative		Second derivative	
Problem 2	Max	L_2	Max	L_2	Max	L_2
0.1 / 0.05	1.96	1.97	1.37	1.74	1.09	1.58
0.05 / 0.025	1.98	1.99	1.71	1.87	1.02	1.72
0.025 / 0.0125	2.00	2.00	1.85	1.94	1.00	1.80
0.0125 / 0.00625	2.00	2.00	1.92	1.97	1.00	1.84
Problem 7						
0.1 / 0.05	1.95	1.95	1.25	1.69	0.95	1.45
0.05 / 0.025	1.98	1.99	1.61	1.85	0.97	1.64
0.025 / 0.0125	2.00	2.00	1.77	1.92	0.98	1.74
0.0125 / 0.00625	2.00	2.00	1.86	1.96	0.99	1.80
Problem 10						
0.1 / 0.05	1.95	1.97	1.38	1.74	1.02	1.56
0.05 / 0.025	1.99	2.00	1.70	1.87	0.99	1.70
0.025 / 0.0125	2.00	2.00	1.83	1.93	0.99	1.78
0.0125 / 0.00625	2.00	2.00	1.89	1.97	1.00	1.83

Figure 2.3. Computed discretization error estimates for problem 2, 7 and 10 using the quadratic boundary approximation.

None of the other test problems had convergence rates significantly slower than the ones listed above. Improved rates were observed for problems 4, 5 and 9 where $R_5 \approx R_6 \approx 2$. Figure 2.4 gives the rate of convergence of the solution at the points $P \in R_h^*$ for problem 7.

Problem 7	Solution		First derivative		Second derivative	
$P \in R^*_h$	Max	L_2	Max	L_2	Max	L_2
0.1 / 0.05	2.56	2.53	1.25	1.40	0.95	1.11
0.05 / 0.025	2.76	2.74	1.61	1.63	0.97	1.32
0.025 / 0.0125	2.85	2.85	1.77	1.78	0.98	1.40
0.0125 / 0.00625	2.91	2.92	1.86	1.87	0.99	1.45

Figure 2.4 Problem 7. Convergence of boundary layer R^*_h using the quadratic boundary approximation.

Improved convergence at the points $P \in R^*$ was again observed for problems 4 ($R_5 \approx R_6 \approx 2$), 5 and 9, where $R_1 \approx R_2 \approx 4$, $R_3 \approx R_4 \approx 3$, and $R_5 \approx R_6 \approx 2$.

Based on the numerical evidence, the discretization error estimates given in Figure 2.5 are believed to be correct for smooth functions.

	Solution		First derivatives		Second derivatives	
Domain	Max	L_2	Max	L_2	Max	L_2
R	2	2*	2	2*	1	$\sqrt{3.5}$
R^*_h	3	3*	2	2	1	1.5

Figure 2.5 Asymptotic behavior of discretization errors.

The entries in Figure 2.5 marked with a star correspond to estimates that are sharp in Theorem 2.1. In addition, the estimate for the maximum error in the solution over R is almost sharp. A specific numerical

calculation strongly suggested that the factor $|\log h^{-1}|^{\frac{1}{2}}$ in Theorem 2.1 can be removed.

The discretization error in the second derivatives is of particular interest. The second derivatives represent important physical quantities both in the theory of elasticity and in fluid applications. The calculations clearly indicate that the maximum error is proportional to h. Theorem 2.1 states that the L_2 error is bounded by $h^{3/2}$, and Zlámal [1967] suggests that this is in fact sharp. However, the numerical results indicate that the rate is $h^{\sqrt{3.5}}$. A possible explanation of this rate would be a boundary layer of thickness $O(h^{-1/2})$ with errors of $O(h^{3/2})$ and interior errors of $O(h^2)$. A special calculation showed that the error in the interior indeed behaved like $O(h^2)$. The value of R_6 is 1.86, 1.86 and 1.84 for problems 2, 7 and 10 when using $h_1 = \frac{1}{200}$ and $h_2 = \frac{1}{256}$ consistent with the value $\sqrt{3.5} = 1.8708$. (Convergence is slow since the next term in the error expansion (ch^2) is only slightly smaller.)

The very regular behavior of the truncation error when using quadratic extrapolation near the boundary suggests that Richardson extrapolation will be quite effective. Figure 2.6 displays the corresponding results of problem 7 after one Richardson extrapolation. (Using h and $h/2$).

Problem 7	Solution		First derivative		Second derivative	
$P \in R$	Max	L_2	Max	L_2	Max	L_2
0.1 / 0.05	3.02	3.79	1.90	2.90	0.97	1.96
0.05 / 0.025	2.99	3.82	1.98	2.94	0.99	1.97
0.025 / 0.0125	3.99	3.94	1.99	2.97	1.00	1.99
$P \in R^*_h$						
0.1 / 0.05	3.02	3.48	1.90	2.45	0.97	1.47
0.05 / 0.025	2.99	3.47	1.98	2.47	0.99	1.47
0.025 / 0.0125	2.89	3.48	1.99	2.48	1.00	1.49

Figure 2.6 Problem 7. One Richardson extrapolation.

It should be pointed out that a precise knowledge of the asymptotic discretization error is vital when doing Richardson extrapolation. Gupta [1979] reports that the maximum error in the solution of problem 5 decreases from 0.07 with $h = 0.05$ to 0.05 when extrapolating using $h = 0.1$ and $h = 0.05$. His extrapolation was based on an expansion with a leading term $h^{3/2}$, if the correct extrapolation is performed the error decreases from 0.07 to 0.002.

Next, consider the discretization errors when using the 13-point formula in combination with cubic extrapolation near the boundary. Problems 1 and 3 are solved exactly by this approximation. Figure 2.7 shows the results for problems 2, 7 and 10, while Figure 2.8 shows the rate of convergence near the boundary for problem 7.

	Solution		First derivative		Second derivative	
Problem 2	Max	L_2	Max	L_2	Max	L_2
0.1 / 0.05	3.04	3.13	2.42	2.85	2.02	2.56
0.05 / 0.025	3.17	3.31	2.69	2.99	2.03	2.68
0.025 / 0.0125	3.23	3.45	2.58	2.90	1.82	2.59
Problem 7						
0.1 / 0.05	2.74	2.27	2.14	2.07	1.61	2.03
0.05 / 0.025	1.41	1.46	1.94	1.76	1.92	2.12
0.025 / 0.0125	1.74	1.72	1.84	1.80	2.08	2.07
Problem 10						
0.1 / 0.05	2.59	2.47	2.24	2.21	2.02	2.23
0.05 / 0.025	1.54	1.60	2.25	1.90	2.02	2.20
0.025 / 0.0125	1.79	1.75	1.93	1.87	2.01	2.10

Figure 2.7 Computed discretization error estimates for problems 2, 7 and 10 using the cubic boundary approximation.

Problem 7	Solution		First derivative		Second derivative	
$P \varepsilon R_h^*$	Max	L_2	Max	L_2	Max	L_2
0.1/0.05	3.28	3.34	2.14	2.20	1.61	1.69
0.05/0.025	3.55	3.63	2.24	2.45	1.92	1.93
0.025/0.0125	3.66	3.75	2.47	2.61	2.08	2.00

Figure 2.8 Problem 7. Boundary layer R_h^* using cubic boundary approximation.

The behavior is not as consistent as in the previous case, making Richardson extrapolation less attractive. Notice that the error in the second derivative near the boundary, is converging at a much better rate than in the first case. The theory in section 2.2 indicates that this method should be $O(h^2)$ in both solution, first and second derivative. The rate of convergence of the solution at the boundary R_h^* would be

3.5 if Lemma 2.2 applied.

The 4th order approximation (2.9) with boundary formulas taken from Zurmühl [1957] was tried with $h = \frac{1}{10}$, $\frac{1}{15}$ and $\frac{1}{20}$. Again problems 1 and 3 are solved exactly by the approximation. Results for problem 7 are given in Figure 2.9.

Problem 7	Solution		First derivative		Second derivative	
$P \in R$	Max	L_2	Max	L_2	Max	L_2
0.10/0.0667	5.41	5.58	4.41	5.00	4.03	4.61
0.0667/0.05	5.87	5.07	4.87	4.91	4.29	4.77
$P \in R_h^*$						
0.10/0.0667	5.77	5.80	4.41	4.69	4.03	4.23
0.0667/0.05	6.05	6.13	4.87	5.02	4.29	4.43

Figure 2.9 Problem 7. A 4th order accurate method.

Despite fairly large variations in the computed rates of convergence, all errors are reduced by a factor of four or more indicating that the method is fourth order accurate. If Lemma 2.1 and 2.2 were valid in this case, then the results for the boundary layer R_h^* would be 5.5. Perhaps more important, the complicated approximation formulas used near the boundary may not be necessary. An approximation of $O(h)$ would probably suffice in order to get accurate function values, while $O(h^2)$ may be necessary to get $O(h^4)$ accuracy also for the second derivatives.

The results given above reflect the strong smoothing properties of the biharmonic operator. The errors in the interior behave nicely even for higher discrete derivatives. Close to the boundary the situation is much more complex, a boundary approximation which is 3 orders less accurate than the interior approximation is sufficient in order to obtain

good convergence for the solution and the first derivative, if the approximation is 2 orders less accurate, then the second derivatives also converge at an optimal rate.

In order to compare the relative accuracy of the four schemes discussed in this section, Figure 2.10 displays the actual error (and the various centered differences of the error) for problem 7.
Figure 2.10 indicates that:

1) The cubic boundary extrapolation produces more accurate results than the quadratic approximation on a given grid.

ii) Richardson extrapolation is very effective when using the quadratic boundary approximation.

iii) On smooth problems like the ones considered here, the fourth order method produces excellent results.

Before closing this section, the importance of a good set of test problems should be mentioned. This study revealed many cases where one or more terms in the (unknown) error expansions dropped out for a given problem. In particular problems 4, 5 and 9 are rather special and give atypical results. (These problems have been considered by several authors in the past.) In many applications the problems will be less smooth than the above test problems. In such cases a fine grid calculation with a second order accurate method is likely to be more satisfactory than a high order, coarse grid calculation.

Problem 7		Solution		First derivative		Second derivative	
	Quadratic bnd.approx.	Max	L_2	Max	L_2	Max	L_2
I	$P \in R$	$5.94 \cdot 10^{-6}$	$2.30 \cdot 10^{-6}$	$7.28 \cdot 10^{-5}$	$1.51 \cdot 10^{-5}$	$3.16 \cdot 10^{-3}$	$2.43 \cdot 10^{-4}$
	$P \in R_h^*$	$1.00 \cdot 10^{-6}$	$7.54 \cdot 10^{-7}$	$7.28 \cdot 10^{-5}$	$5.53 \cdot 10^{-5}$	$3.16 \cdot 10^{-3}$	$1.33 \cdot 10^{-3}$
	One Richardson extrap.						
II	$P \in R$	$1.46 \cdot 10^{-8}$	$7.04 \cdot 10^{-10}$	$7.00 \cdot 10^{-7}$	$3.15 \cdot 10^{-8}$	$1.59 \cdot 10^{-4}$	$4.95 \cdot 10^{-6}$
	$P \in R_h^*$	$1.46 \cdot 10^{-8}$	$3.52 \cdot 10^{-9}$	$7.00 \cdot 10^{-7}$	$1.81 \cdot 10^{-7}$	$1.59 \cdot 10^{-4}$	$4.15 \cdot 10^{-5}$
III	Cubic bnd. approx.						
	$P \in R$	$8.69 \cdot 10^{-7}$	$3.37 \cdot 10^{-7}$	$4.06 \cdot 10^{-6}$	$1.81 \cdot 10^{-6}$	$1.34 \cdot 10^{-4}$	$1.98 \cdot 10^{-5}$
	$P \in R_h^*$	$3.59 \cdot 10^{-8}$	$2.08 \cdot 10^{-8}$	$2.12 \cdot 10^{-6}$	$1.24 \cdot 10^{-6}$	$1.34 \cdot 10^{-4}$	$9.08 \cdot 10^{-5}$
	4th order method						
IV	$P \in R$	$1.49 \cdot 10^{-8}$	$6.09 \cdot 10^{-9}$	$1.49 \cdot 10^{-7}$	$4.32 \cdot 10^{-8}$	$4.01 \cdot 10^{-6}$	$6.49 \cdot 10^{-7}$
	$P \in R_h^*$	$1.15 \cdot 10^{-8}$	$7.65 \cdot 10^{-9}$	$1.49 \cdot 10^{-7}$	$1.10 \cdot 10^{-7}$	$4.01 \cdot 10^{-6}$	$2.39 \cdot 10^{-6}$

Figure 2.10. Problem 7. Comparison of accuracy.
$h = 0.0125$ in I and III, $h = 0.0125$ and
$h = 0.00625$ was used in II, while
$h = 0.05$ in IV .

2.4 A brief survey of other methods.

A large number of papers proposing numerical algorithms for the approximate solution of the continuous problem (1.1) have appeared in the literature. The rapid development of increasingly faster computers in the last two decades has made it feasible to actually solve finite difference approximations to the biharmonic equation proposed and theoretically investigated as early as 1928 in the important paper by Courant, Friedrichs and Lewy.

Today, there is a considerable interest not only in the various discrete approximations of a given continuous problem, but also in the computational complexity of the discrete problem itself. The solution of the discrete Poisson equation is a good illustration. In the last fifteen years many efficient numerical methods have been developed. (Hockney [1965], Buneman [1969], Buzbee, Golub and Nielson [1970], Bank and Rose [1977] and Schröder, Trottenberg and Witsch [1978].) When solving the problem on an N by N grid $O(N^2)$ arithmetic operations and $O(N^2)$ storage is needed. A method having this complexity is said to be optimal. (Actual computer implementations often make use of the fast Fourier transform or the idea of cyclic reduction resulting in nearly optimal methods having an operation count of $O(N^2 \log N)$.) These methods can all be viewed as efficient computer implementations of the separation of variables technique.

However, separation of variables cannot be applied to the biharmonic problem (1.1). The methods proposed in earlier papers, for the solution of the discrete problem that arises when using the 13-point stencil have not been optimal. The main result of the next chapter is to show that a numerical method of optimal complexity does exist, even though the matrix

corresponds to a nonseparable problem. (It seems however, that optimal numerical methods for this type of problems do require the use of an iterative process.)

The methods that have been proposed, for solving the linear system of equations

$$Ax = b$$

derived from the 13-point stencil can roughly be classified as follows:

i) Iterative methods working on the matrix A.

ii) Direct methods working on the matrix A.

iii) Iterative methods based on reducing the biharmonic problem to a coupled system of two second order equations involving the Laplace operator.

iv) Direct methods taking advantage of the fact that A can be split into $L^2 + V$, where L is the discrete Laplace operator and V has low rank.

The first approach i) can be found in many early papers on the subject; Parter [1959] and Conte and Dames [1960]. A more recent paper using a strongly implicit scheme is Jacobs [1973]. The main disadvantage of approach i) is related to the fact that A has condition number proportional to N^4 resulting in slow convergence of the iterative techniques. (Munksgaard [1980] reports that more than 500 conjugate gradient iterations are required already for $N = 32$.)

Approach ii) has recently received more attention due to a better understanding of sparse methods for Gauss elimination. The theoretical complexity of a direct method using nested dissection is $O(N^3)$ arithmetic operations and $O(N^2 \log N)$ storage locations. Nested dissection

and other sparse matrix methods for the problem were studied by Sherman [1975]. His results indicate that the constants in the above estimates are quite large and that a regular band solver ($O(N^4)$ work, $O(N^3)$ storage) is competitive even when the number of unknowns approach one thousand. Bauer and Reiss [1972] proposed a block elimination scheme, while Gupta and Manohar [1979] used a band solver. Both these methods require a prohibitive amount of storage if N is large and they have a typical running time proportional to N^4, unacceptable for fine grid calculations.

The third and fourth approach are essentially two different ways of looking at the same underlying problem. A method based on iii) above was introduced by Smith [1968]. It had a running time of $O(N^3)$. This was later improved to $O(N^{5/2})$ by Smith [1970], [1973], Ehrlich [1971]. (See also Ehrlich and Gupta [1975].) A drawback is the need to estimate iteration parameters. Recently Vajteršic [1979] presented a more efficient implementation of these ideas, but the complexity of the method remained $O(N^{5/2})$.

The last appraoch iv) was pioneered by Golub [1971] and a refined implementation is given by Buzbee and Dorr [1974]. This implementation, which is a direct method, requires $O(N^3)$ arithmetic operations. Despite being an $O(N^3)$ method it proved very competitive with the $O(N^{5/2})$ methods on realistic problems because those methods have an actual cost of $c\,N^{5/2}$ with c substantially larger than the constant in the $O(N^3)$ estimates.

Based on the above results Sameh, Chen and Kuck [1976] concluded that the solution of the first biharmonic problem was an order or magnitude more difficult than the solution of Poisson's equation even on parallel

computers. The results of this thesis show that the problems have the same complexity.

There are many alternative ways, not using finite differences, for obtaining an approximate solution to the biharmonic equation. A few will be mentioned here.

i) Finite element methods.

An extensive literature exists. The methods of solution are most often sparse Gaussian elimination. Recent contributions proposing alternative ways of solving the resulting linear equations include Axelsson and Munksgaard [1979] and Glowinski and Pironneau [1979].

ii) Least squares methods.

Methods of this type are often called "point matching methods" in the engineering literature while the name "method of particular solutions" sometimes is used by numerical analysts. This approach can be very effective for special problems. References include McLaurin [1968], Sigillito [1976] and Rektorys [1979].

iii) Integral equation methods.

A large number of papers have appeared and the theoretical foundation is well understood. (See the references given in Chapter I). A few recent papers are Katsikadelis [1977], Richter [1977] and Cristiansen and Hougaard [1978].

iv) Methods using Fourier series expansions.

A few papers construct the solution of the first biharmonic problem in a rectangular region using infinite Fourier expansions. References to work in this direction include Aronszajn, Brown and Butcher [1973], Vaughan [1974] and Rahman and Usmani [1977].

v) Methods using mathematical programming.

Linear programming techniques have been used by Cannon and Cecchi [1966], [1967] and Dessi and Manca [1976] while Distéfano [1971] reports on the use of a continuous dynamic programming technique.

CHAPTER III

AN $O(N^2)$ METHOD FOR THE SOLUTION OF THE FIRST BIHARMONIC PROBLEM

Consider the Dirichlet problem for the biharmonic operator in a rectangle R.

$$\begin{aligned} \Delta^2 u(x,y) &= f(x,y) \qquad (x,y) \in R \\ u(x,y) &= g(x,y) \qquad (x,y) \in \partial R \\ u_n(x,y) &= h(x,y) \qquad (x,y) \in \partial R \end{aligned} \tag{3.1}$$

Here u_n denotes the normal derivative of u with respect to the exterior normal.

A new and more efficient solution technique will be described for the case when the above system is discretized using the standard 13-point stencil combined with quadratic extrapolation at the boundary. It will be shown that by using this method, the solution of the discrete problem on an N by N grid can be computed in $O(N^2)$ arithmetic operations. This is an order of magnitude faster than earlier methods. In addition, the storage requirement is also significantly reduced compared to previously published algorithms.

The theory in this chapter does not uniquely define a numerical algorithm. In fact, it will become clear that there are several ways of implementing $O(N^2 \log N)$ methods as well as an even faster direct $O(N^2 \log N)$ method requiring $O(N^3)$ operations in a preprocessing stage. It should be pointed out that the $\log N$ term only arises when doing a fast Fourier transform that can be associated with solving Poisson's equation on the given grid. Several methods for solving Poisson's

equation in only $O(N^2)$ operations are known. Work in this direction has been reported by Banks [1978] and Detyna [1979], while Swarztrauber [1977] gives an $O(N^2 \log\log N)$ method.

It is possible to use one of these methods as a subprogram in the algorithms described in this chapter, and this would result in a fast biharmonic solver requiring only $O(N^2)$ arithmetic operations to achieve a prescribed accuracy.

There are at least four reasons for keeping the discrete Fourier transform (and therefore the $\log N$ term) in this description of the new method.

i) The theory becomes clearer and more coherent.

ii) The $O(N^2)$ methods for Poisson's equation are still research codes of limited availability and several have problems with numerical instabilities.

iii) The fast Fourier transform is a more widely used computational tool. Very efficient codes already exist and hardware implementations are likely to exist on many computer systems in the future. The constant in front of the $N^2 \log N$ term is also quite small compared to the constant in front of the N^2 term. Under these circumstances the $\log N$ penalty may be of little significance in actual computation.

iv) The fast Fourier transform is used anyway in a different part of the algorithm. (It only makes an $O(N \log N)$ contribution to the operation count in this part, so a slow Fourier transform would not change the asymptotic efficiency).

A more detailed analysis of several variants of the algorithm with precise descriptions of actual computer implementations including storage

requirements and operation counts is given in Chapter IV.

In this section the structure and properties of the discrete matrix problem corresponding to (3.1) will be analyzed. Since the basic method of solution is closely related to this structure, the analysis will be carried out as a constructive derivation of the algorithm.

Assume that the rectangle R is discretized using a grid with M uniformly spaced interior gridpoints in the x-direction and similarly N points in the y-direction. The resulting linear system of MN equations is

$$Au_h = b$$

with $(u_h)_{ij}$, $i = 1,2,\ldots M$, $j = 1,2,\ldots N$ denoting the discrete approximation to the continuous solution $u(x,y)$ at the coordinate $(i\Delta x, j\Delta y)$. The vector b is given by

$$b_{ij} = (\Delta y)^4 f(i\Delta x, j\Delta y) + \ell_{ij}$$

where the sparse vector ℓ is a linear combination of the boundary data corresponding to the quadratic boundary approximation discussed in Chapter II.

In order to discuss the efficient numerical solution of this system some notation is needed.

Let

$$\delta = \Delta y/\Delta x$$

and define the two matrices

$$R_N = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & . & & & \\ & . & . & . & & \\ & & . & . & . & \\ & & & . & . & . \\ & & & & -1 & 2 \end{bmatrix}_{N\times N} \qquad T_N = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & . & \\ & & & 1 \end{bmatrix}_{N\times N}$$

Let I_N denote an $N \times N$ identity matrix. The matrix A can be written as

$$A = \left[\delta^2(I_N \otimes R_M) + (R_N \otimes I_M)\right]^2 + 2(T_N \otimes I_M) + 2\delta^4(I_N \otimes T_M) \qquad (3.2)$$

Standard tensor product notation is used, i.e.

$$C = (D_N \otimes E_M)$$

denotes the block $NM \times NM$ *matrix (with blocksize* M)

$$C = \begin{bmatrix} d_{11}E & . & . & . & d_{1N}E \\ . & d_{22}E & & & . \\ . & & . & . & . \\ . & & & . & . \\ d_{N1}E & . & . & . & d_{NN}E \end{bmatrix}_{NM\times NM}$$

Note that the matrix

$$L = \delta^2(I_N \otimes R_M) + (R_N \otimes I_M) \qquad (3.3)$$

is nothing but the matrix that results when solving Poissons equation on the same grid using the standard 5-point difference approximation to the Laplace operator.

Consider the $N \times N$ symmetric matrix

$$Q_N = \{q_{ij}\} = \sqrt{\frac{2}{N+1}} \{\sin \frac{ij\pi}{N+1}\} \quad . \tag{3.4}$$

It is easy to show that the vectors q_i $i = 1,2...N$ are the normalized eigenvectors of R_N and that

$$\begin{aligned} &Q_N R_N Q_N = \Lambda_N \\ &Q_N = Q_N^T = Q_N^{-1} \\ &\Lambda_N = \text{diag}(\lambda_j) \\ &\lambda_j = 2(1-\cos \frac{j\pi}{N+1}) \quad j = 1,2...N \quad . \end{aligned} \tag{3.5}$$

Notice that the operation of computing $y = Q_N x$ for a given vector x of length N is just a real sine-transform of x. It can therefore be carried out in $O(N\log N)$ arithmetic operations using the fast Fourier transform. For this discussion the $MN \times MN$ permutation matrix P, $P^T P = I$ defined by the relation

$$P(D_N \otimes E_M)P^T = (E_M \otimes D_N)$$

is also needed. If P acts on the vector u_h it will reorder the unknowns by columns (vertically), instead of by rows (horizontally). It is clear that this involves no arithmetic operations.

The matrix T_N is of rank 2 and it can be written

$$T_N = U_N U_N^T$$

where

$$U_N = \begin{bmatrix} 1 & 0 \\ 0 & . \\ . & . \\ . & . \\ . & 0 \\ 0 & 1 \end{bmatrix}_{N\times 2} \quad .$$

Define

$$\hat{Q} = (I_N \otimes Q_M)$$

and

$$K_N = Q_N U_N \quad .$$

Now consider

$$\begin{aligned} P\,\hat{Q}\,A\,\hat{Q}\,P^T &= \delta^4(\Lambda_M^2 \otimes I_N) + (I_M \otimes R_N^2) + 2\,\delta^2(\Lambda_M \otimes R_N) \\ &\quad + 2(I_M \otimes U_N U_N^T) + 2\,\delta^4(K_M K_M^T \otimes I_N) \qquad (3.6) \\ &\equiv \hat{S} + 2\,\delta^4(K_M K_M^T \otimes I_N) \quad . \end{aligned}$$

This defines the block diagonal matrix

$$\hat{S} = \begin{bmatrix} S_1 & & \bigcirc \\ & \ddots & \\ \bigcirc & & S_M \end{bmatrix}_{NM \times NM}$$

Explicitly written out, block number k of $\hat{S}$ has the following penta-diagonal structure:

$$S_k = \begin{bmatrix} 7+4\delta^2\lambda_k+\delta^4\lambda_k^2, & -4-2\delta^2\lambda_k, & 1 & & & & \\ -4-2\delta^2\lambda_k, & 6+4\delta^2\lambda_k+\delta^4\lambda_k^2, & \ddots & \ddots & & & \\ 1 & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & 1 & \\ & & \ddots & \ddots & 6+4\delta^2\lambda_k+\delta^4\lambda_k^2, & -4-2\delta^2\lambda_k & \\ & & & 1 & -4-2\delta^2\lambda_k, & 7+4\delta^2\lambda_k+\delta^4\lambda_k^2 & \end{bmatrix}_{N \times N} \qquad (3.7)$$

After these transformations the problem is reduced to

$$\left[\hat{S} + 2\delta^4(K_M K_M^T \otimes I_N)\right] v = c \qquad (3.8)$$

where the transformed variables $v = P\hat{Q}u_h$ and $c = P\hat{Q}b$ have been introduced. It is important to notice that the matrix $(K_M K_M^T \otimes I_N)$ has rank $2N$ only.

The following generalization of the Sherman-Morrison formula is well known (Dahlquist and Björck, [1974]).

Let $E \in R^{n\times n}$ be nonsingular, $V \in R^{n\times p}$, and $W \in R^{n\times p}$. Then

$$(E + VW^T)^{-1} = E^{-1} - E^{-1}V(I + W^T E^{-1}V)^{-1}W^T E^{-1} \qquad (3.9)$$

provided that the $p\times p$ matrix $(I + W^T E^{-1}V)$ is nonsingular.

Applying this formula to equation (3.8) makes it possible to write down an explicit expression for the solution u_h (returning to the original variables).

$$u_h = (I_N \otimes Q_M)P^T \left[I - 2\delta^4 \hat{S}^{-1}(K_M \otimes I_N)B^{-1}(K_M^T \otimes I_N)\right]\hat{S}^{-1}P(I_N \otimes Q_M)b \qquad (3.10)$$

where B is the $2N \times 2N$ matrix

$$B = I + 2\delta^4(K_M^T \otimes I_N)\,\hat{S}^{-1}(K_M \otimes I_N) \quad . \qquad (3.11)$$

By looking at the different matrices in (3.10), performing the operations from right to left it is clear that:

i) $(I_N \otimes Q_M)$ requires $O(NM\log M)$ operations. (N fast Fourier transforms of length M).

ii) P requires no operations. (A permutation only).

iii) $\hat{S}^{-1}$ requires $O(NM)$ operations. (M pentadiagonal systems of size N).

iv) $(K_M^T \otimes I_N)$ requires $O(NM)$ operations. ($K_M^T \otimes I_N$ has sparse simple structure).

v) B^{-1} requires ? operations. (B has not been analyzed yet).

In this way, the design of an efficient method has been reduced to the fast solution of a linear system with coefficient matrix B. In the following a careful study of the matrix B is made and a method of solving such linear systems in no more than $O(NM)$ arithmetic operations is obtained.

In order to do this, taking advantage of the structure in B, the matrix can be written:

$$B = I + 2\delta^4 (K_M^T \otimes I_N)\, \hat{S}^{-1}\, (K_M \otimes I_N)$$

$$= I + 2\delta^4 \begin{bmatrix} q_{11}I_N & \cdots & q_{M1}I_N \\ \\ q_{1M}I_N & \cdots & q_{MM}I_N \end{bmatrix} \begin{bmatrix} S_1^{-1} & & \\ & \ddots & \\ & & S_M^{-1} \end{bmatrix} \begin{bmatrix} q_{11}I_N & & q_{1M}I_N \\ \vdots & & \vdots \\ q_{M1}I_N & & q_{MM}I_N \end{bmatrix}$$

$$= I + 2\delta^4 \begin{bmatrix} \sum_{k=1}^{M} q_{k1}^2 S_k^{-1} & , & \sum_{k=1}^{M} q_{k1}q_{kM}S_k^{-1} \\ \\ \sum_{k=1}^{M} q_{k1}q_{kM}S_k^{-1} & , & \sum_{k=1}^{M} q_{kM}^2 S_k^{-1} \end{bmatrix}_{2N \times 2N} . \qquad (3.12)$$

Now

$$q_{k1} = \sqrt{\frac{2}{M+1}} \sin \frac{k\pi}{M+1}$$

$$q_{kM} = (-1)^{k+1} q_{k1} \quad .$$

Define

$$S_{odd} = \sum_{k=1,3,5..}^{M} \sin^2 \frac{k\pi}{M+1} S_k^{-1} \tag{3.13}$$

$$S_{even} = \sum_{k=2,4,6..}^{M} \sin^2 \frac{k\pi}{M+1} S_k^{-1} \quad . \tag{3.14}$$

Therefore

$$B = I + \frac{4\delta^4}{M+1} \begin{bmatrix} S_{odd} + S_{even} & S_{odd} - S_{even} \\ S_{odd} - S_{even} & S_{odd} + S_{even} \end{bmatrix}_{2N\times 2N} \tag{3.15}$$

Consider solving a linear system $Bx = d$. Partition $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ into subvectors of length N consistent with the partitioning of B. By adding and subtracting equations this system splits into two linear systems each of size $N \times N$:

$$(I + \frac{8\delta^4}{M+1} S_{odd})\ (x_1 + x_2) = d_1 + d_2 \tag{3.16}$$

$$(I + \frac{8\delta^4}{M+1} S_{even})(x_1 - x_2) = d_1 - d_2 \quad . \tag{3.17}$$

It will later be shown that these problems can be split further into four symmetric positive definite matrix problems each of size $N/2$. ($N/2$ will be used to denote both $(N+1)/2$ and $(N-1)/2$ if N is odd, the actual value being clear from context.) However, since all known practical direct methods for solving a general dense linear system of equations of order N require $O(N^3)$ arithmetic operations, it is natural to study possible iterative methods. (There exist certain direct methods for very special classes of matrices, for example Toeplitz matrices, with a lower operation count, but the matrices under consideration do not seem to belong to any such class). Notice also that the matrices S_{odd} and S_{even} are

defined by rather complicated relations. In fact, it would require $O(N^2M)$ arithmetic operations to generate all the explicit matrix elements. Since all the pentadiagonal blocks S_k of $\hat{S}$ are symmetric and positive definite, it follows that both S_{odd} and S_{even} have the same properties.

A very attractive iterative scheme for the solution of a symmetric positive definite linear system

$$Ax = b$$

is the conjugate gradient method. From an arbitrary initial vector x_0 the method generates a sequence of approximations $\{x_n\}$ to the solution x defined by

$$x_{n+1} = x_n + \alpha_n p_n \quad , \alpha_n = \frac{(r_n, r_n)}{(Ap_n, p_n)}$$

$$p_{n+1} = r_{n+1} + \beta_n p_n \, , \beta_n = \frac{(r_{n+1}, r_{n+1})}{(r_n, r_n)} \tag{3.18}$$

where $r_n = b - Ax_n$ and $p_0 = r_0$.

The method is due to Hestenes and Stiefel [1952]. A good description of the method and some of its properties can be found in Luenberger [1973]. The iteration does not require knowledge of the matrix elements, since only matrix vector products are needed. It is clear from the structure of S_{odd} and S_{even} (3.13, 3.14) that a matrix vector product can be computed by solving $M/2$ of the pentadiagonal systems S_k. The cost of a matrix vector product is therefore $O(NM)$ arithmetic operations. The number of iterations required to achieve a given accuracy when solving a symmetric positive definite system of linear equations $Ax = b$ using conjugate gradients, is in general proportional to $(\mu_{max}/\mu_{min})^{\frac{1}{2}}$ where

$\{\mu_i\}_{i=1}^{N}$ is the spectrum of A. It should be pointed out that special distributions, in particular clusters of eigenvalues, will lead to a considerably faster rate of convergence. (Kaniel [1966], Stewart [1975], Cline [1976], Jennings [1977], and Greenbaum [1979].)

It can be shown using the results of Appendix II that the largest eigenvalues of the matrices given in (3.16, 3.17) both are proportional to M. A direct application of conjugate gradients to those linear systems will therefore require at most $O(NM^{3/2})$ arithmetic operations. This is of the same order of magnitude as the method known in the literature as "The coupled equation approach" described and studied by Smith [1968, 1970, 1973], Ehrlich [1971, 1972, 1973], Greenspan and Schultz [1972], McLaurin [1974] and Gupta [1975].

The iterative techniques proposed in these papers all require various acceleration parameters to be estimated. In addition, each iteration amounts to solving two full Poisson problems. The actual use of these methods have been restricted to rectangular regions since the handling of the boundary and the need to compute normal derivatives there, is quite complicated for more general domains.

The use of a conjugate gradient iteration has several advantages over the iterative methods proposed earlier. The method requires no estimation of iteration parameters and it takes advantage of the spectral distribution of the linear operator in an optimal way. Thus a more careful study of the spectrum (see Appendix II) reveals that it clusters around 1 and that the large eigenvalues behave like cM/i, $i = 1,2,\ldots$. It can be shown that the conjugate gradient method converges in $O(M^{1/3})$ iterations if the arithmetic is exact. Unfortunately inexact arithmetic makes the actual number of iterations (using 3.18) behave more like $O(M^{1/2})$.

The use of a few quasi-Newton updates (see Chapter IV) or selective orthogonalization (Parlett [1980]) as part of the conjugate gradient procedure, provides a remedy for this problem.

With an operation count of $O(NM^{4/3})$, this algorithm is faster than those mentioned above. In practice, even with a standard conjugate gradient implementation, the method is substantially faster than previous algorithms.

The main purpose of this chapter is however, to show that linear systems defined by the matrix B (3.11) can be solved using only O(NM) arithmetic operations.

Suppose, instead of applying the conjugate gradient method directly to a linear system $Tx = b$, that it is possible to split T such that

$$T = \tilde{T} - R$$

where $\tilde{T}$ is symmetric positive definite. Assume in addition that it is easy to solve linear systems with the matrix $\tilde{T}$. In such a case, the conjugate gradient method can be used with a preconditioning matrix $\tilde{T}$ corresponding to the above splitting of T. This can equivalently be viewed as applying ordinary conjugate gradient iteration to the transformed system

$$\tilde{T}^{-\frac{1}{2}} T \tilde{T}^{-\frac{1}{2}} y = c$$

but working with the original variables $x = \tilde{T}^{-\frac{1}{2}} y$ and $b = \tilde{T}^{\frac{1}{2}} c$. The number of iterations needed in order to achieve a given accuracy is therefore in general again proportional to the ratio $(\mu_{max}/\mu_{min})^{\frac{1}{2}}$, but $\{\mu_i\}_{i=1}^{N}$ are now the eigenvalues of the matrix

$$K = \tilde{T}^{-1} T = I - \tilde{T}^{-1} R \quad .$$

(K is of course similar to the symmetric matrix $\tilde{T}^{-\frac{1}{2}} T \tilde{T}^{-\frac{1}{2}}$ given above).

A good analysis of this technique, including numerical algorithms, is given in Concus, Golub and O'Leary [1976]. If $\tilde{T}^{-1}$ is an approximate inverse of T then the convergence rate will be much improved. Two different effects can contribute in this process.

i) The ratio μ_{max}/μ_{min} is often substantially reduced when considering K instead of the original matrix T.

ii) Equally important is the fact that K often will have clusters of eigenvalues. Typically, K will have only $p << N$ eigenvalues appreciably different from 1. The number of iterations required for convergence will then be similar to the number required for a problem of dimension p with the corresponding spectrum.

The next few pages will describe how to find a splitting of the present problem (3.16, 3.17) that has both of the above properties.

Write

$$\begin{aligned} S_k &= \tilde{S}_k + 2U_N U_N^T \\ &= \tilde{S}_k + 2e_1 e_1^T + 2e_N e_N^T \ . \end{aligned} \tag{3.20}$$

Comparing with (3.5) and (3.7) it is clear that

$$\tilde{S}_k = (\delta^2 \lambda_k I_N + R_N)^2 \tag{3.21}$$

and therefore all the matrices $\tilde{S}_k$, $k = 1,2,...M$ have the same set of eigenvectors represented by the matrix Q_N (3.4).

$$Q_N \tilde{S}_k Q_N = \Psi_k$$

$$\Psi_k = \text{diag}(\Psi_{kj}) \tag{3.22}$$

$$\Psi_{kj} = (\delta^2 \lambda_k + \lambda_j)^2 \ , \ j = 1,2,\ldots N$$

(Recall that λ_k is defined in (3.5) and that this definition depends implicitly on the range k is running over.)

In the following, let

$$T_i = (I + \frac{8\delta^4}{M+1} \sum_{k=i,i+2,..}^{M} \sin^2 \frac{k\pi}{M+1} S_k^{-1}) \quad i = 1,2 \tag{3.23}$$

represent the matrix in both linear systems (3.16) and (3.17). The notation $\sum_{k=i,i+2}^{M}$ indicates that the summation extends over odd or even k (depending on i) up to M. Let $\tilde{T}_i$ represent the matrix

$$\tilde{T}_i = (I + \frac{8\delta^4}{M+1} \sum_{k=i,i+2,..}^{M} \sin^2 \frac{k\pi}{M+1} \tilde{S}_k^{-1}) \quad i = 1,2 \tag{3.24}$$

where $\tilde{S}_k^{-1}$ has replaced S_k^{-1} in (3.23). T_i and $\tilde{T}_i$ can both be viewed as discrete approximations to certain boundary integral operators relating the solution of (3.1) to the solution of the separable problem where Δu is specified on two opposite parts of the boundary instead of u_n. In particular, $\tilde{T}_i$ corresponds to a separable operator. There is a close correspondence between these operators (in the rectangular case) and the integral operator A defined in Glowinski-Pironneau [1979].

When using conjugate gradients to solve a linear system involving the matrix T_i, consider a preconditioning corresponding to the following splitting:

$$T_i = \tilde{T}_i - (\tilde{T}_i - T_i) \ . \tag{3.25}$$

Observe that

$$\tilde{T}_i = Q_N(I + \frac{8\delta^4}{M+1} \sum_{k=i,i+2..}^{M} \sin^2 \frac{k\pi}{M+1} \Psi_k^{-1})Q_N \qquad (3.26)$$

$$= Q_N D_i Q_N \quad . \qquad i = 1,2$$

The matrix D_i defined in (3.26) is diagonal and can be computed in $O(NM)$ operations. The solution of linear systems involving $\tilde{T}_i$ can therefore be performed in $O(N\log N)$ operations once the matrix D_i has been computed and stored.

Lemma 3.1

The matrices T_i, $\tilde{T}_i$ and $\tilde{T}_i - T_i$ are symmetric, T_i and $\tilde{T}_i$ positive definite and $\tilde{T}_i - T_i$ positive semi-definite.

Proof:

The statement about T_i and $\tilde{T}_i$ follows trivially from the definitions (3.5), (3.7), (3.23) and (3.24).

Consider the matrix $\tilde{T}_i - T_i$

$$\tilde{T}_i - T_i = \frac{8\delta^4}{M+1} \sum_{k=i,i+2..}^{M} \sin^2 \frac{k\pi}{M+1} (\tilde{S}_k^{-1} - S_k^{-1}) \quad .$$

But, using (3.20) and the Sherman-Morrison formula results in

$$S_k^{-1} = \tilde{S}_k^{-1}(I - 2\, U_N(I_2 + 2U_N^T\tilde{S}_k^{-1}U_N)^{-1}\, U_N^T\, \tilde{S}_k^{-1}) \quad .$$

Thus

$$\tilde{S}_k^{-1} - S_k^{-1} = 2\, \tilde{S}_k^{-1}\, U_N(I_2 + 2U_N^T\tilde{S}_k^{-1}U_N)^{-1}U_N^T\tilde{S}_k^{-1} \quad .$$

This matrix is clearly positive semi-definite. □

Lemma 3.2

Let $K_i = \tilde{T}_i^{-1}\, T_i$ and let $\{\mu_{ik}\}_{\nu=1}^{N}$ be the spectrum of K_i, then

$0 < \mu_{ik} \leq 1$, $1 \leq k \leq N$, $i = 1,2$.

Proof:

It follows from Lemma 3.1 and the fact that $\tilde{T}_i^{-1} T_i$ is similar to $\tilde{T}_i^{-\frac{1}{2}} T_i \tilde{T}_i^{-\frac{1}{2}}$ that $\mu_{ik} > 0$.

Let x be any vector, $x^T x = 1$. Using Lemma 3.1 it follows that

$$x^T \tilde{T}_i^{-\frac{1}{2}} (\tilde{T}_i - T_i) \tilde{T}_i^{-\frac{1}{2}} x \geq 0$$

which implies

$$x^T \tilde{T}_i^{-\frac{1}{2}} T_i \tilde{T}_i^{-\frac{1}{2}} x \leq 1$$

and

$$\mu_{ik} \leq 1, \quad k = 1,2,..N \quad i = 1,2 \quad . \square$$

In order to prove that the preconditioned conjugate gradient method proposed above converges at a rate independent of N, a theorem giving more precise knowledge than Lemma 3.2 about the spectrum of K_i is needed.

The next theorem describing a matrix decomposition of T_i leads to new variants of the algorithm as well as better knowledge about the eigenvalues $\{\mu_{ik}\}_{k=1}^{N}$.

Theorem 3.1

For $i \in \{1,2,3,...\}$, define $i_r = 2i - \delta_{r1}$, $\delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$.

Let

$$\beta_M = 8\delta^4/(M+1) \quad \text{and} \quad \beta_N = 8/(N+1)$$

and

$$\alpha_k^{rs} = 1 + \beta_s \sum_{j=r,r+2,..}^{s} \sin^2 \frac{j\pi}{(s+1)} \psi_{kj}^{-1} \quad .$$

Let P_N be the permutation matrix that permutes a vector $x \in R^N$ odd-even, i.e., if x has components $(x_1,x_2,x_3,...x_N)$, then Px has

components $(x_1, x_3, \ldots x_{N/2}, x_2, x_4, \ldots x_N)$. Then

$$P_N D_i^{-\frac{1}{2}} Q_N T_i Q_N D_i^{-\frac{1}{2}} P_N^T = I_N - \begin{bmatrix} (C^{i1})^T C^{i1} & & \bigcirc \\ & & \\ \bigcirc & & (C^{i2})^T C^{i2} \end{bmatrix}_{N\times N} \qquad i = 1,2$$

where C^{rs} is the $M/2$ by $N/2$ matrix with components

$$c_{ij}^{rs} = \frac{\sqrt{\beta_N \beta_M}\, \sin \frac{i_r \pi}{M+1} \sin \frac{j_s \pi}{N+1}}{\sqrt{\alpha_{i_r}^{sN} \alpha_{j_s}^{rM}}\, \psi_{i_r j_s}}$$

$r = 1,2$

$s = 1,2$

$i = 1,2,\ldots M/2, \frac{M+3-2r}{2}$ if M odd

$j = 1,2,\ldots N/2, \frac{N+3-2s}{2}$ if N odd

Proof:

See Appendix I. □

First observe that the two matrix problems associated with $T_i (i=1,2)$ have been split into a total of four smaller problems. This reduces the required computer storage, but this fact is even more important in the design of a direct method. The reduction into four subproblems is a direct consequence of the symmetry of the biharmonic operator on the rectangle R. A second important observation is that the elements c_{ij}^{rs} can be easily computed after some initial computation of the quantities that appear in the above formula. This requires only $O(NM)$ operations and $O(N) + O(M)$ storage and provides an alternative to the implicit definition of T_i given in (3.23).

It is now possible to prove a stronger result than Lemma 3.2. Let $\{\sigma_k^{ij}\}_{k=1}^{N/2}$ be the singular values of C^{ij}, and let $\{\mu_{ik}\}_{k=1}^{N}$ be the eigenvalues of $K_i = \tilde{T}_i^{-1} T_i$. Clearly, from Theorem 3.1, there is a one to one

correspondence between μ and σ given by

$$\mu = 1 - \sigma^2 \tag{3.27}$$

where the subscripts and superscripts have been dropped for notational convenience.

First consider the case where $M = N$ and $\delta = 1$. In this case C^{11} and C^{22} are square and symmetric, while $C^{12} = (C^{21})^T$ is almost square. This case is slightly simpler to analyze and will be considered first.

Theorem 3.2

Assume $N = M$ and $\delta = 1$. Let $\{\sigma_i\}_{i=1}^{N/2}$ be the singular values of one of the matrices C^{rs} defined in Theorem 3.1. Then

$$0 \leq \sigma_i < 0.8$$

independent of N.

Proof:

See Appendix II. Explicit expressions for the matrix elements c_{ij}^{rs} are derived in the limiting case $N \to \infty$. A simple Gershgorin estimate can be applied in this case to give an upper bound for the largest singular value σ_i. The largest singular value σ_i and the actual computed upper bound are shown in Figure 3.1 for N ranging from 1 to 2047.

Computations show that σ_{max} always belong to C^{11}. A block Lanzcos code written by Underwood [1975] was used to compute the eigenvalues in Figure 3.1. The theoretical Gershgorin bound when N tends to infinity is also indicated. Although sufficient for this theory, the figure indicates that the bounds are not very sharp for realistic values of N.

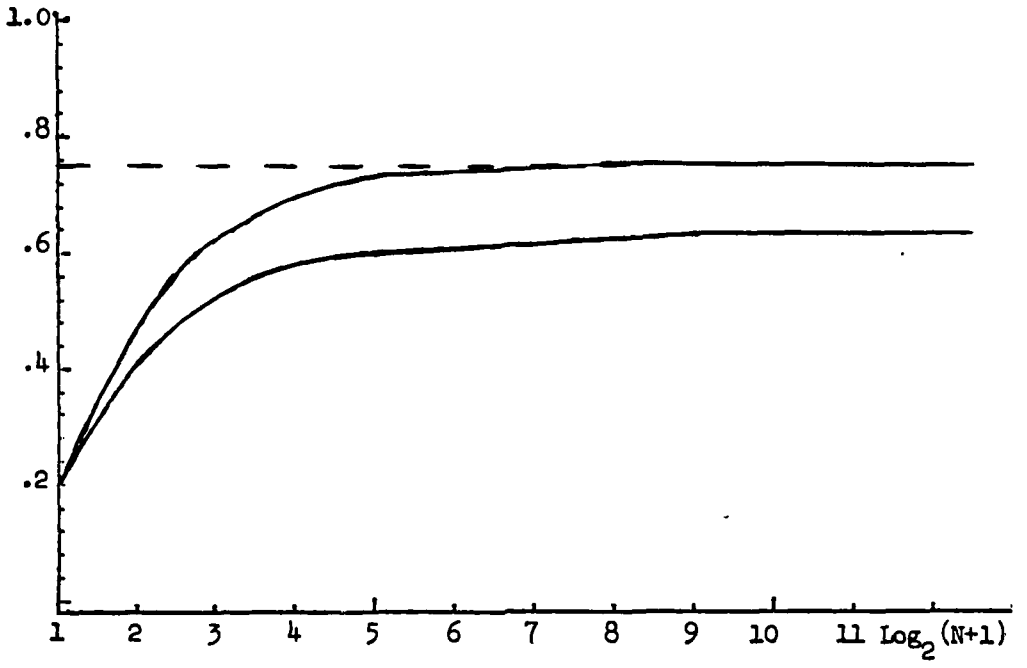


Figure 3.1. The largest singular value as a function of $\log_2(N+1)$ (below), compared with the corresponding Gershgorin bound (above).

Theorem 3.2 shows that $0 \leq \sigma_i < 0.8$, the next theorem implies that the singular values σ_i cluster at zero.

Theorem 3.3

$$\sum_{i=1}^{N/2} \sigma_i < \ln N \quad \text{if } \sigma_i \text{ belong to } C^{11} \text{ or } C^{22}$$

$$\sum_{i=1}^{N/2} \sigma_i^2 < \ln N \quad \text{if } \sigma_i \text{ belong to } C^{12} \text{ or } C^{21} \ .$$

Proof:

See Appendix II. □

Remark: The weaker statement when σ_i belong to C^{12} or C^{21} is stated because the proof technique becomes simpler. See the comments in Appendix II. Notice that Theorem 3.3 implies that only $O(\log N)$ of the σ_i's are outside any given neighborhood of zero. With this information about the $\{\sigma_i\}_{i=1}^{N/2}$ it is possible to give some estimates of the rate of convergence of the proposed conjugate gradient iteration.

Theorem 3.4

If the conjugate gradient algorithm is used to solve the linear system $T_i x = b$ with the splitting $T_i = \tilde{T}_i - (\tilde{T}_i - T_i)$, then the initial error will be reduced by a factor ε after at most

$$k = \ln(\frac{2}{\varepsilon})$$

iterations.

Proof:

Let $\mu_1 < \mu \leq \ldots \leq \mu_N$. It is well known from the standard theory of the conjugate gradient method that

$$\varepsilon \leq 1/T_k \left(\frac{\mu_N+\mu_1}{\mu_N-\mu_1}\right)$$

where T_k is the k'th Chebychev polynomial of the first kind. $T_k(x) = \cosh(k \cosh^{-1}x)$ for $x > 1$. Therefore

$$k \leq \frac{\cosh^{-1}(1/\varepsilon)}{\cosh^{-1}(\frac{\mu_N+\mu_1}{\mu_N-\mu_1})} \quad .$$

Using $\cosh^{-1}(\frac{1}{\varepsilon}) < \ln(\frac{2}{\varepsilon})$, $\mu_1 > 1 - .8^2 = .36$, and $\cosh^{-1}(\frac{1+.36}{1-.36}) > 1$ gives the desired result. □

This theorem establishes convergence to any prescribed accuracy in a constant number of iterations independent of N. Since each iteration takes $O(N^2)$ arithmetic operations the description of an $O(N^2 \log N)$ algorithm for the first biharmonic problem is complete. If the accuracy is required to increase with increasing N as N^{-p} for a fixed p, then $O(\log N)$ iterations are required and the overall asymptotic operation count remains unchanged. (In order to be consistent with a decreasing truncation error $p = 2$).

However, under the above assumptions the use of an $O(N^2)$ Poisson solver will not make the overall algorithm any faster if the solution on the final grid is computed directly. In order to have an $O(N^2)$ method under these assumptions, it is necessary to compute the solution on a sequence of grids, reducing the error by a fixed amount on each grid. (The total work on all the coarser grids will only be $O(N^2)$.)

For practical computations ($N \leq 2047$) the use of the computed spectral radius $\sigma_1 = .6343$ for $N = 2047$ (See Figure 3.1) strengthens the above theorem to

$$k \leq \tfrac{1}{2} \ln \left(\frac{2}{\varepsilon}\right) .$$

As an illustration taking $\varepsilon = 10^{-10}$, this estimate gives $k \leq 12$.

The above theorems show that the conjugate gradient iteration converges at a very fast linear rate. The next theorem complements this by showing that asymptotically the rate of convergence is in fact superlinear.

A sequence $\{e_k\}_{k=0}^{\infty}$ converges R-superlinearly to zero if and only if $\limsup_{k\to\infty} \|e_k\|^{1/k} = 0$. An excellent reference discussing the convergence of iterative processes is Ortega and Reinboldt [1970].

Theorem 3.5

The conjugate gradient method defined in Theorem 3.4 has a R-superlinear rate of convergence.

Proof:

Using the optimality property of the conjugate gradient iteration

$$\|e_k\| \equiv (c_k)^k \|e_0\| \leq \max_{\mu \in \{\mu_i\}_{i=1}^N} \prod_{j=1}^{k} \left|\frac{\mu_j - \mu}{\mu_j}\right| \|e_0\|$$

where $\|e_k\|$ is the error in the appropriate norm at iteration k. Let the set $\{\mu_i\}_{i=1}^N$ be ordered such that $\mu_i \leq \mu_{i+1}\ \forall\ i$. Then

$$\|e_k\| \leq \max_{\mu \in \{\mu_i\}_{i=k+1}^N} \prod_{j=1}^{k} \left|\frac{\mu_j - \mu}{\mu_j}\right| \|e_0\|$$

$$\leq \max_{\sigma \in \{\sigma_i\}_{i=k+1}^N} \prod_{j=1}^{k} \frac{\sigma_j^2 - \sigma^2}{1-\sigma_j^2} \|e_0\|$$

$$\leq \prod_{j=1}^{k} \frac{\sigma_j^2}{1-\sigma_j^2} \|e_0\| \ .$$

Using the arithmetic-geometric mean inequality, Theorem 3.3 and the fact that $\sigma_j < 1\ \forall j$ gives

$$\|e_k\| \leq \left[\frac{1}{k} \sum_{j=1}^{k} \frac{\sigma_j^2}{1-\sigma_j^2}\right]^k \|e_0\|$$

$$\leq \left[\frac{1}{k} \frac{\ln N}{1-\sigma_1^2}\right]^k \|e_0\|$$

This inequality shows that the constant

$$c_k \leq \frac{1}{k} \frac{\ln N}{1-\sigma_1^2}$$

tends to zero as k increases for fixed N.

However, since the concept of R-superlinear convergence is most meaningful in the case of an infinite number of iterations and the conjugate gradient method has finite termination on finite dimensional problems, consider the limiting case as $N \to \infty$. Theorem 3.3 implies that

$$\lim_{k\to\infty} \sigma_k = 0$$

and therefore

$$\lim_{k\to\infty} c_k = \lim_{k\to\infty} \left[\frac{1}{k} \sum_{j=1}^{k} \frac{\sigma_j^2}{1-\sigma_j^2} \right] = 0 \quad . \quad \square$$

Finally, consider the case where $N \neq M$. Without loss of generality assume $\delta = (M+1)/(N+1)$. (δ was defined to be $\Delta y/\Delta x$). This restriction corresponds to a linear scaling of one of the independent variables in the differential equation. Let $C^{rs}_{M\times N}$ denote the $M/2$ by $N/2$ matrix C^{rs} derived from an $M\times N$ grid. The following relations hold:

$$\begin{aligned} C^{11}_{M\times N} &= (C^{11}_{N\times M})^T \\ C^{12}_{M\times N} &= (C^{21}_{N\times M})^T \\ C^{22}_{M\times N} &= (C^{22}_{N\times M})^T \end{aligned} \tag{3.28}$$

as can be seen from the definition of C^{rs} in Theorem 3.1.

Using the same technique as in Appendix II, this time working with the singular values of all four matrices, it can be shown that the largest singular value is smaller than what it is in the case of a square grid with

max(N,M) gridpoints in each coordinate direction. (An alternative approach is to show that each element c_{ij}^{rs} decreases if M or N is reduced. The spectral radius of a non-negative matrix is at least as large as that of a principal minor, and it increases if an element of the matrix increases. This, together with the claim about the element c_{ij}^{rs} above, leads to the desired conclusion.) (3.28) shows that M and N enter the problem in a completely symmetric way and it is sufficient to consider the case $N \leq M$. (This choice saves both storage and arithmetic operations in the conjugate gradient iteration.) The largest singular value will again belong to the matrix C^{11}. Figure 3.2 shows the computed value of σ_{max} for various values of $N \leq M$. The corresponding Gershgorin bounds obtained using $\sigma_{max} \leq (\|C^{rs}\|_1 \|C^{rs}\|_\infty)^{\frac{1}{2}}$ are shown in the same figure for $N > M$ (the case $N = M$ is in Figure 3.1). Finally, in Figure 3.3, the largest singular value in each of the four different matrices C^{rs} are computed for a few values of M and N.

It is felt that the computed spectral data combined with the theory in this chapter provide a good foundation for using the proposed conjugate gradient iteration in practical computer codes for the biharmonic equation.

GERSHGORIN BOUND

MAX SINGULAR VALUE: M \ N	3	7	15	31	63	127	255
3	.41	.52	.55	.55	.55	.55	.56
7	.46	.53	.66	.67	.67	.67	.67
15	.48	.55	.58	.72	.72	.73	.73
31	.48	.55	.59	.60	.74	.75	.75
63	.48	.56	.59	.61	.61	.75	.75
127	.48	.56	.59	.61	.62	.62	.76
255	.48	.56	.59	.61	.62	.62	.63

Figure 3.2. Max singular value and Gershgorin bound.

	N	3	7	15	31	63	127	255
	C^{11}	.41	.53	.58	.60	.61	.62	.63
M=N	C^{12}	.18	.32	.41	.48	.52	.55	.56
δ=1	C^{21}	.18	.32	.41	.48	.52	.55	.56
	C^{22}	.11	.27	.39	.46	.51	.54	.56
	C^{11}	.46	.55	.59	.61	.62	.62	
M=2N+1	C^{12}	.23	.36	.44	.49	.53	.55	
δ=2	C^{21}	.22	.35	.44	.49	.53	.55	
	C^{22}	.17	.31	.41	.48	.52	.55	
	C^{11}	.48	.55	.59	.61	.62		
M=4N+3	C^{12}	.25	.37	.48	.50	.53		
δ=4	C^{21}	.24	.36	.45	.50	.52		
	C^{22}	.18	.33	.42	.48	.48		

Figure 3.3. Max singular value for each matrix C^{rs} .

CHAPTER IV

COMPUTER ALGORITHMS

This chapter will describe a few computer algorithms implementing the ideas in the previous chapter. The more general equation

$$\begin{aligned} \Delta^2 u(x,y) + \alpha\, \Delta\, u(x,y) + \beta\, u(x,y) &= f(x,y) \quad (x,y)\in R \\ u(x,y) &= g(x,y) \quad (x,y)\in \partial R \\ u_n(x,y) &= h(x,y) \quad (x,y)\in \partial R \end{aligned} \tag{4.1}$$

will be considered. Efficient methods for solving a sequence of such problems, as well as the performance of the proposed algorithms on vector and parallel computers will be described. Numerical results showing the stability of the numerical process with respect to roundoff errors can be found in section 4.6. Section 4.7 discusses how to solve the discrete approximation when the cubic extrapolation defined in (2.4), is used near the boundary. An efficient numerical method for the solution of the first biharmonic boundary value problem in a circular disk is given in section 4.8, while section 4.9 indicates how problems in different geometries may be handled using conformal mapping.

4.1 An algorithm using Fourier transform and penta-diagonal linear systems.

All the algorithms to be presented here are based on the theory developed in Chapter III. Quite a few arithmetic operations as well as storage locations, can be saved by paying close attention to the way various expressions are related. Although these aspects are important in order to produce an efficient code, some details are omitted in this presentation. The algorithms are stated in a form closely corresponding to actual computer programs. It is convenient (but not necessary) to assume

that both N and M are odd.

A few definitions are needed before the algorithms can be stated:

$$\tilde{\alpha} \equiv (\Delta y)^2 \alpha$$

$$\tilde{\beta} \equiv (\Delta y)^4 \beta$$

$$\mu_k \equiv 2 + 4\,\delta^2 \sin^2\left(\tfrac{1}{2}\frac{k\pi}{M+1}\right) \qquad k = 1,2,\ldots M .$$

$$\tilde{Q}_N \equiv \sqrt{8(N+1)}\, Q_N$$

$$S_k \equiv \text{Pentadiag}\left[1, \tilde{\alpha} - 2\mu_k, 2 + \tilde{\beta} + \mu_k(\mu_k - \tilde{\alpha}), \tilde{\alpha} - 2\mu_k, 1\right] + T_N \ .$$

The notation (aside from new definitions) is consistent with the notation introduced in the beginning of Chapter III. The unnormalized transform $\tilde{Q}_N$ is used in order to conform with the fast Fourier transform package written by Swarztrauber [1978] and used in the computation reported in this thesis.

Let the vector f_{ij} $(i = 1,2,\ldots M,\ j = 1,2,..N)$ represent the discrete right hand side function in (4.1) and let the sparse vector ℓ_{ij} contain the contribution from the boundary data g and h to the right hand side of the discrete linear system. The subvector $(f_{i1}, f_{i2}..f_{iN})$ will be written $f_{i\bullet}$, while $(f_{1j}, f_{2j}, \ldots f_{Mj})$ is written $f_{\bullet j}$. Also let x,y,z,s and p represent five (work) vectors each of length N.

Algorithm 4.1

1. For $j=1,2,..N$:

$$f_{\bullet j} := (\Delta y)^4 f_{\bullet j} + \ell_{\bullet j}$$

$$f_{\bullet j} := \tilde{Q}_M f_{\bullet j} \ .$$

2. $x := 0$.

3. For $i=1,3,5,\ldots M$:

$$y := S_i^{-1} f_{i\bullet}$$

$$x := x + \left(\frac{\delta^4}{(M+1)^2} \sin \frac{i\pi}{M+1}\right) y$$

$$f_{i\bullet} := \frac{1}{8(M+1)} y$$

4. Solve a linear system of equations $Cy = x$ using conjugate gradients preconditioned by the matrix H.

A. A matrix multiply $p := Cs$ is defined by:

$$p := s$$

For $i=1,3,5,\ldots M$:

$$z := S_i^{-1} s$$

$$p := p + \left(\frac{8\delta^4}{M+1} \sin^2 \frac{i\pi}{M+1}\right) z \quad .$$

B. A preconditioning step $p := Hs$ is defined by:

$$p := \tilde{Q}_N s$$

$$p := D^{-1} p$$

$$p := \tilde{Q}_N p \quad .$$

The diagonal matrix D has (precomputed) elements defined by:

$$d_j = 8(N+1)\left(1 + \frac{8\delta^4}{M+1} \sum_{i=1,3,\ldots} \frac{\sin^2 \frac{i\pi}{M+1}}{(4\sin^2 \tfrac{1}{2} \frac{j\pi}{N+1} + \mu_i - 2)(4\sin^2 \tfrac{1}{2} \frac{j\pi}{N+1} + \mu_i - 2 - \tilde{\alpha}) + \tilde{\beta}}\right)$$

5. For i=1,3,5,...M :

$$x := S_i^{-1} y$$

$$f_{i\bullet} := f_{i\bullet} - (\sin \tfrac{i\pi}{M+1})x \ .$$

6. Repeat steps 2,3,4 and 5 with i running over even integers instead of odd. (i=2,4,6,...M everywhere.)

7. For j=1,2,...N :

$$f_{\bullet j} := \tilde{Q}_M \, f_{\bullet j} \ .$$

8. Stop. The discrete solution of equation (4.1) is now stored in the vector f_{ij}.

Remark.

Trigonometric functions needed in the above algorithm should be precomputed and saved in an array of size 2(N+M). The two diagonal matrices D used in 4, must also be precomputed requiring an additional 2N storage locations. Notice that only a few vectors of length N are needed in step 4; the big vector f_{ij} is never accessed.

The conjugate gradient iteration will converge in a small number of iterations as long as the corresponding linear system is positive definite. That is certainly the case as long as $\alpha \leq 0$ and $\beta \geq 0$. If the system is indefinite a routine like SYMMLQ by Paige and Saunders [1975] can be substituted. (Alternatives are a least squares formulation or the algorithm given in section 4.3.)

The performance of algorithm 4.1 depends on an efficient solution of

the pentadiagonal linear systems $S_i y = x$. Consider first the case where S_i is positive definite. Taking advantage of the special structure, the factorization of this matrix requires 3N operations (2N multiplications and N divides plus 3N adds) and 2N words of storage. (The combination of a multiplication/addition or a divide/addition will be considered one arithmetic operation.) The solution process after the factorization has been completed, takes 4N operations. (4N mult/add only.) One possibility is to save all the factorizations when they are computed the first time. This would apparently require $O(N^2)$ storage. However, a much better alternative is to observe that the matrix elements in the factored form of S_i converge. The rate of convergence increases with increasing i. This process can be analyzed and the final (converged) values of the elements can be computed directly from the matrix S_i. Figure 4.1 illustrates the savings obtained when the factorization is computed with an accuracy of 10^{-16}. (The gain is larger if a less accurate factorization is acceptable.) The current computer implementation therefore recomputes this factorization every time it is needed. Alternatively the necessary information could be stored, requiring much less storage than what is often used in similar codes today. This technique can also be used advantageously in fast Poisson solvers.

N	50	100	200	400	800	1600
(#op)/1000 using full factorization. (= $3N^2/1000$)	7.5	30	120	480	1920	7680
(#op)/1000 using convergence. (≡ p/1000)	3.6	8.6	20.0	45.2	100.6	220.7
p/(NlnN)	18.5	18.7	18.9	18.9	18.8	18.7

Figure 4.1. The total factorization cost of all the matrices $S_i (i=1,2,...N)$ for the case $\alpha = \beta = 0$.

If S_i is indefinite and α or β is zero, then S_i can be written

$$S_i = R_i^+ R_i^- + 2\, T_N \qquad (T_N \equiv e_1 e_1^T + e_N e_N^T)$$

where R_i is tridiagonal. R_i^+ is positive definite while R_i^- can be viewed as representing a two term recursion with characteristic roots inside the unit disk. Linear systems involving R_i^- can therefore be solved in a stable way by using a marching procedure. Combining this with the Sherman-Morrison formula, results in a stable algorithm requiring $9N^2$ operations and $2N$ storage locations in order to solve N indefinite systems each of size N. Frequently, only a few systems are indefinite, resulting in an operation count between $4N^2$ and $9N^2$.

If S_i is indefinite with both α and β nonzero then band Gauss elimination can be used, but it is likely that algorithm 4.2 or 4.3 would be a better choice in this case.

The sine-transform (represented by the matrix $\tilde{Q}_N$) can be computed using a complex fast Fourier transform of length $N/2$. The operation count depends on the prime factors of N. A complex fast Fourier transform of length N can be computed using $\frac{1}{2}N\log_2 N$ complex multiplications if $N = 2^k$ and with no more than $N \sum_{i=1}^{k} (n_i - 1)$ complex multiplications if $N = \prod_{i=1}^{k} n_i$ (Henrici [1979]).

Assume for simplicity that the number of real operations required to form $y = \tilde{Q}_N x$ is $N\log_2 N$. This corresponds to the multiplications required when $N = 2^k - 1$, see Temperton [1979], [1980] for more detailed operation counts. With these assumptions, the total operation count for algorithm 4.1 (ignoring lower order terms) is

$$NM(2\log_2 M + 5k + 12) \qquad (4.2)$$

where k is the average number of conjugate gradient iterations for the two linear systems in step 4.

Figure 4.2 shows the execution time of this algorithm on an IBM 370/168 using the FORTRAN H (Opt = 3) compiler. What is important is of course the general behavior of the algorithm rather than the specific times. Average running times based on problems 2,3 and 7 on five different grids, are given. The conjugate gradient iteration in step 4 was stopped when the 2-norm of the residual fell below the specified tolerance TOL. This results roughly in a comparable accuracy of the final solution to the discrete problem. The numbers include all preprocessing and do not represent a fully optimized code. The execution times have been split into the time required by the Fourier transform in step 1 and 7 (FFT) and the remainder (SOLV). The gridsizes $N = 121$ and $N = 243$ result in an unfavorable prime factor of 61 when doing the Fourier transform. The execution time increases somewhat slower with N than indicated by (4.2), reflecting lower order contributions omitted in (4.2).

N	63	121	127	243	255
FFT	136	2386	472	7518	1862
SOLV (TOL = 10^{-5})	408	1434	1355	4774	4953
SOLV (TOL = 10^{-10})	557	1905	1784	6773	6931
TOTAL (TOL = 10^{-5})	544	3820	1827	12292	6815

Figure 4.2 Execution time in milliseconds for algorithm 4.1.

The average number of conjugate gradient iterations is given in Figure 4.3.

N	63	121	127	243	255
TOL = 10^{-5}	5	5	5	5	5
TOL = 10^{-10}	7	8	8	9	9

Figure 4.3 Average number of conjugate gradient iterations required. (Problems 2,3 & 7)

4.2 An algorithm based on Fourier transformations.

The complete decomposition of the discrete problem into four subproblems as described in Chapter III, becomes clearer if sine-transforms are applied in both coordinate directions. In this case it is necessary to solve linear systems of the form:

$$\tilde{Q}_N \, S_i \, \tilde{Q}_N \, x = y \quad . \tag{4.3}$$

It follows from Theorem 3.1 that this system decouples (odd-even) into two systems of half the size. The subalgorithm PENTF (i,r,y,x) solves the odd-numbered equations if $r = 1$ and the even-numbered if $r = 2$. Here y and x are unscaled vectors of length $N/2$ containing the appropriate components from (4.3). (For simplicity both N and M are assumed odd, $(N+1)/2$ and $(N-1)/2$ are both written $N/2$, the actual value being clear from the context.)

This algorithm can be derived from the decomposition given in Chapter III. In the following description define $k \equiv 2j - 1$ if $r = 1$ and $k \equiv 2j$ if $r = 2$.

Subalgorithm PENTF (i,r,y,x).

1. Define a (work) vector w of length N/2 by:

$$w_j := \frac{\sin \frac{k\pi}{N+1}}{(4\sin^2 \tfrac{1}{2}\frac{k\pi}{N+1} + \mu_i - 2)(4\sin^2 \tfrac{1}{2}\frac{k\pi}{N+1} + \mu_i - 2 - \tilde{\alpha}) + \tilde{\beta}}$$

2. Let

$$c_1 := \frac{8}{N+1} \sum_{j=1}^{N/2} \sin \frac{k\pi}{N+1} w_j$$

$$c_2 := \frac{8}{N+1} \sum_{j=1}^{N/2} y_j w_j$$

$$a := \frac{c_2}{1+c_1} \quad .$$

3. The solution is now given by:

$$x_j := \left(\frac{y_j}{\sin \frac{k\pi}{N+1}} - a\right) w_j \quad .$$

Remark.

The quantity c_1 can be precomputed at the expense of of $2M$ storage locations. Also the divide in step 3 can be changed into a multiply by storing $1/\sin \frac{k\pi}{N+1}$.

The full algorithm can now be st[illegible]. Let x,y,z,s and p be five (work) vectors of length N/2 and [illegible] k as above.

Algorithm 4.2

1. For j=1,2,...N :

$$f_{\bullet j} := (\Delta y)^4 f_{\bullet j} + \ell_{\bullet j}$$

$$f_{\bullet j} := \tilde{Q}_M f_{\bullet j} \quad .$$

For $i=1,2,\ldots M$:

$f_{i\cdot} := \tilde{Q}_N f_{i\cdot}$

2. Let $x := 0$, $r := 1$

3. For $i=1,3,5,\ldots M$:

$z_j := f_{ik} \qquad (j=1,2,\ldots N/2)$

PENTF (i,r,z,z)

$$x := x + \left(\frac{1}{8(N+1)} \frac{\delta^4}{(M+1)^2} \sin \frac{i\pi}{M+1}\right) z$$

$$f_{ik} := \frac{1}{64(N+1)(M+1)} z_j \qquad (j=1,2,\ldots N/2)$$

4. Solve a linear system of equations $Cy = x$ using conjugate gradients preconditioned by the matrix D^{-1}.

A. A matrix-vector multiply $p := Cs$ is defined by

$p := s$

For $i=1,3,\ldots M$:

PENTF(i,r,s,z)

$$p := p + \left(\frac{8\delta^4}{M+1} \sin^2 \frac{i\pi}{M+1}\right) z \quad .$$

B. A preconditioning step $p := D^{-1}s$ is defined by:

$p_j := d_k^{-1} s_j \qquad (j=1,2,\ldots N/2)$

where the diagonal matrix D is defined in algorithm 4.1 step 4. (Note that the summation in that definition extends over odd or even i.)

5. For $i=1,3,5,\ldots M$:

PENTF(i,r,y,x)

$$f_{ik} := f_{ik} - \sin \frac{i\pi}{M+1} x_j \qquad (j=1,2,\ldots N/2)$$

6. Repeat steps 2,3,4 and 5 with i running over even integers instead of odd. ($i = 2,4,6,\ldots M$ everywhere).

7. Let $x := 0$, $r := 2$, repeat steps 3,4,5 and 6.

8. For $i=1,2,\ldots M$:

$$f_{i\bullet} := \tilde{Q}_N f_{i\bullet} \ .$$

For $j=1,2,\ldots N$:

$$f_{\bullet j} := \tilde{Q}_M f_{\bullet j} \ .$$

9. Stop. The discrete solution of equation (4.1) is now stored in the vector f_{ij}.

Remark:

This algorithm has no restrictions on the parameters α and β, but rapid convergence of the conjugate gradient algorithm is only guaranteed when the corresponding linear systems are definite. (See the discussion in section 4.1).

Algorithm PENTF(i,j,x,y) requires 5N operations if x and y are N-vectors. Using the same assumptions as in section 4.1, the (asymptotic) operation count for algorithm 4.2 is:

$$NM(2\log_2 NM + 6k + 14) \qquad (4.4)$$

where k is the average number of conjugate gradient iterations for the four systems. Results for this algorithm corresponding to figure 4.2 are given in figure 4.4, and the average number of conjugate gradient iterations is given in figure 4.5.

N	63	121	127	243	255
FFT	292	4792	1034	15442	4316
SOLV (TOL=10^{-5})	367	1280	1394	4890	5392
SOLV (TOL=10^{-10})	501	1688	1869	6601	7272
Total (TOL=10^{-5})	659	6072	2428	20332	9708

Figure 4.4 Execution time in milliseconds for algorithm 4.2.

N	63	121	127	243	255
TOL = 10^{-5}	3	3	3	3	3
TOL = 10^{-10}	5	5	5	5	5

Figure 4.5 Average number of conjugate gradient iterations required. (Problems 2,3 & 7)

Algorithm 4.2 is a very good alternative to algorithm 4.1, in particular if a sequence of problems are being solved and it is possible to work with Fourier transformed variables. If this is the case the cost of the Fourier transforms may be ignored. This is certainly the case when computing discrete approximations to the eigenvalues and eigenfunctions of the continuous problem. It should also be pointed out that symmetries in a given problem may greatly reduce the computational work, since the vector x in step 4 then often will be zero. (Resulting in a trivial problem.) In this respect the numerical algorithm can be viewed as an efficient numerical implementation of the decomposition of the space of solutions to the first biharmonic problem into four orthogonal subspaces. (See Aronszajn, Brown and Butcher [1973], Vaughan [1974] and Fichera [1966]). This property is further discussed in the beginning of Chapter V.

4.3 An efficient direct method.

Algorithm 4.2 can also serve as a basis for a direct method. Instead of using conjugate gradients to solve the four linear systems of order $N/2$, in step 4, the systems can be solved by using symmetric Gauss elimination. If an indefinite symmetric solver is used, (Aasen [1971], Bunch and Parlett [1971]) then all nonsingular discrete analogs of (4.1) can be solved.

Algorithm 4.3

This algorithm is identical to algorithm 4.2 except for step 4.

4. A. Generate the elements of the matrix C.

   B. Factor the matrix C using a symmetric factorization.

   C. Solve the linear system $Cy = x$ using the computed factorization of C.

If a sequence of problems with the same parameters α and β, are solved on the same grid then steps 4A and 4B need not be repeated. The computer implementation of this algorithm uses the routines DPPFA and DPPSL or DSPFA and DSPSL from LINPACK (Dongarra, Bunch, Moler and Stewart [1979]).

There remains to show how to generate the matrix elements. The columns of C are determined by repeated use of step 4A in algorithm 4.2, choosing the vectors s as the columns of the identity matrix. Exploiting symmetry and the fact that s is sparse, all four matrices can be generated using $MN^2/4 + O(MN)$ arithmetic operations. The same operationcount results if the matrices C^{rs} given in theorem 3.1, are generated and then multiplied together. In fact, the two processes are equivalent. The factorization cost in step 4B, for all four matrices is

$N^3/12 + O(N^2)$. The extra storage needed in order to save all four factorizations, in steps 4A and 4B, is only $N^2/2$. The leading term in the operationcount for algorithm 4.3 is

$$\tfrac{1}{4}\, NM(N + \tfrac{1}{3}\, N^2/M + 8\log_2 NM) + O(NM) \qquad (4.5)$$

for the first right hand side, and

$$NM(2\log_2 NM + N/M + 14) + O(N) + O(M) \qquad (4.6)$$

for additional right hand sides. Figure 4.6 gives the execution time for this algorithm on a VAX-11/780 computer. In order to make comparisons with the previous algorithms easier, the first row in the table has been compared with the corresponding row in figure 4.4 and all other entries are given in approximate IBM 370/168 time using this normalization.

N	63	127	255
VAX FFT	2672	11426	49840
FFT	292	1034	4316
SOLV (4.5)	353	1592	9707
SOLV (4.6)	120	398	1616
Total (4.6)	412	1432	5932

Figure 4.6 Normalized execution time in milliseconds for algorithm 4.3.

Remarks.

i) The algorithms presented in sections 4.1, 4.2 and 4.3 have been stated in order to show the structure and simplicity of a possible computer implementation reflecting

the structure and simplicity of the underlying theory developed in Chapter III.

ii) The algorithms can be improved upon when solving special cases of (4.1). For example, an even better preconditioning matrix can be constructed when $\alpha = \beta = 0$, by incorporating the knowledge from Lemma A2.2. Recall however, that such improvements although important in some applications, only reduces the constant in the operation-count. The complexity of algorithms 4.1 and 4.2 using an $O(N^2)$ Poisson solver, is $O(N^2)$ and this result is optimal.

iii) Algorithms 4.1 and 4.2 will execute faster if it is acceptable to use more storage for intermediate results. If the vector w in subalgorithm PENTF is precomputed and stored (requiring $O(NM)$ storage), then the operation-count of this important subroutine is reduced from $5N$ to $3N$.

iv) A direct method based on algorithm 4.1 has an operation-count of

$$NM(2\log_2 M + N/M + 12) \quad . \tag{4.7}$$

This is somewhat faster than (4.6), but the algorithm is not as general.

4.4 The solution of several problems on the same grid using conjugate gradients.

In this section it is shown that a small modification of algorithms 4.1 and 4.2 can reduce the computational cost when solving a sequence of

problems on the same grid. There are several situations where this can be of interest. When solving very large systems the $O(N^2M)$ preprocessing cost of algorithm 4.3 may be unacceptable. Perhaps more important, the technique described in this section can be used when solving a sequence of problems of the form (4.1), allowing not only f,g, and h, but also the parameters α and β to change.

Consider the following algorithm for solving a symmetric, positive definite linear system $Ax = b$. Given x_0 and H_0 let:

$$x_{n+1} = x_n + \alpha_n p_n \quad , \alpha_n = \frac{(H_{n-1}r_n, r_n)}{(Ap_n, p_n)}$$

$$p_{n+1} = H_n r_{n+1} + \beta_n p_n \ , \ \beta_n = \frac{(H_n r_{n+1}, r_{n+1})}{(H_{n-1}r_n, r_n)} \qquad (4.8)$$

$$H_{n+1} = H_n + U(p_n, Ap_n, H_n)$$

where $r_n = b - Ax_n$, $p_0 = H_0 r_0$ and $H_{-1} = H_0$.

If $H_0 = I$ and $U \equiv 0$ this is the conjugate gradient method (3.18). If $H_0 \neq I$ and $U \equiv 0$ it is preconditioned conjugate gradients with a preconditioning matrix H_0. If $U = U_\beta(s,y,H)$ is any member of the Broyden [1970] β-class of quasi-Newton updates, then in exact arithmetic, this algorithm generates the same sequence $\{x_n\}$ as when $U \equiv 0$. (Nazareth [1979]). In finite precision calculations, the choice $U = U_\beta(s,y,H)$ results in a more stable algorithm when solving problems similar to (3.16) or (3.17) avoiding the characteristic loss of orthogonality (Parlett [1980]), that can affect the rate of convergence of the conjugate gradient method. The process (4.8) can be viewed as a conjugate gradient method with a variable preconditioning matrix (a variable metric)

making the matrix $H_n^{\frac{1}{2}} A H_n^{\frac{1}{2}}$ increasingly more well conditioned. This observation shows that a sequence of problems can be solved more efficiently provided that the information stored in H_n from a previous iteration, is saved.

Since all the updates $U_\beta(s,y,H)$ are equivalent in exact arithmetic, the symmetric rank one update given by

$$U_{SR1}(s,y,H) = \frac{(s-Hy)(s-Hy)^T}{(s-Hy)^T y}, \qquad (4.9)$$

seems to be best suited in this particular situation. At every iteration only one new vector $v_n \equiv (s_n - H_n y_n)$ must be stored. This is an N-vector in algorithm 4.1 and a vector of length N/2 in algorithm 4.2. Algorithm 4.4 outlines how this technique can be used when 2K vectors of length N are available.

Algorithm 4.4.

A. For each new problem apply algorithm 4.1 or 4.2, but use version (4.8) of the conjugate gradient method in step 4. Initially the matrix-vector product $p := H_0 s$ is defined in step 4B, but at step n, $n \leq K$ it will be given by

$$p := H_0 s + \sum_{i=1}^{n} (\gamma_i v_i^T s) v_i \qquad (4.10)$$

where γ_i is the scaling factor from (4.9) and the vector v_i is a stored update.

B. When a total of K conjugate gradient iterations have been performed, (possibly after solving more than one of the problems in the sequence) continue with $U \equiv 0$ and use

$$p := H_0 s + \sum_{i=1}^{K} (\gamma_i v_i^T s) v_i \qquad (4.11)$$

in step 4B.

Only a few updates are needed in order to achieve convergence in one or two iterations and the cost of this procedure does not add to the leading terms in the operationcounts of algorithms 4.1 and 4.2. Algorithm 4.4 can be used advantageously even when the parameters α and β in (4.1) are changing. However, in this case it may sometimes be necessary to restart with H_0 as defined in step 4B of algorithms 4.1 and 4.2. If the BFGS update U_{BFGS} is used, then Nocedal [1979] showed that an interesting, alternative updating strategy is possible.

Algorithm 4.4 in combination with both 4.1 and 4.2, was tried taking K = 5. All 10 test problems were solved on an IBM 370/168 with a stopping tolerance TOL = 10^{-5}. The average time per problem and the average number of conjugate gradient steps needed are given in figures 4.7 and 4.8. For comparison, the same sequence of problems were solved (on a VAX-11/780) using algorithm 4.3. The normalized IBM times are given in figure 4.9 .

N	63	127	255
FFT	137	487	1918
SOLV	281	905	3202
TOTAL	418	1392	5120
#Cg-iterations	2	2.1	2

Figure 4.7 Average execution time (ms) per problem when solving 10 problems with algorithm 4.4/4.1.

N	63	127	255
FFT	290	1034	4308
SOLV	291	1055	4037
TOTAL	581	2089	8345
#Cg-iterations	1.4	1.6	1.7

Figure 4.8 Average execution time (ms) per problem when solving 10 problems with algorithm 4.4/4.2.

N	65	127	255
VAX FFT	2672	11426	49840
FFT	290	1034	4308
SOLV	142	521	2421
TOTAL	432	1555	6729

Figure 4.9 Normalized execution time (ms) per problem when solving 10 problems with algorithm 4.3.

Figure 4.7 and 4.8 show that the number of conjugate gradient iterations decreases significantly. Figure 4.9 and 4.6 show that the preprocessing cost of algorithm 4.3 is also quite acceptable for this problem. It should be noted that the total cost when solving 10 problems using the three different methods are almost equal. In fact, comparing expressions (4.5) and (4.6) with (4.4) indicate that the two methods are equally efficient when

$$p \approx \frac{N}{18\ k} \qquad (4.12)$$

problems are being solved. In (4.12) k is the average number of conjugate gradient iterations required per problem. Taking $N = 255$ and $k = 1.7$, the two methods should be equally efficient when solving 8 problems and this corresponds well with the computational results.

4.5 Algorithms for vector and parallel computers.

Sameh, Chen and Kuck [1976] considered algorithms for Poisson's equation and the biharmonic equation under the assumptions of an "idealized parallel computer" having N^2 or N^3 processors. They concluded that the biharmonic equation was an order of magnitude more difficult than Poisson's equation. This section gives new and improved results, as well as more practical results for the case when a fixed number of processors and/or a vector computer is available. Without loss of generality, the discussion is limited to algorithm 4.2 with $N = M$.

While truly parallel computers having p independent processors with unrestricted communication, exist mostly as theoretical models, vector computers capable of performing arithmetic operations on vector registers, play an increasingly more important role in current large scale scientific computations. An algorithm for the biharmonic equation on such a computer will therefore be considered first. The following simplifying assumptions are made:

i) There are p processors available.

ii) The four arithmetic operations +, -, * and / can be performed by these processors working on vector registers.

iii) An operation (or a timestep) will consist of an addition or subtraction and a multiplication or a divide performed componentwise on vectors of length at most p.

iv) Startup costs including memory and/or data alignment times are ignored.

These assumptions are naturally not fully realistic, but different machine architectures make it difficult to use a more complicated model. An algorithm that performs well under the above assumptions is likely to also be very efficient on pipelined vector-computers like the CRAY-1. Despite having only one processor, this computer performs vectoroperations so efficiently that the model can be used with an effective p larger than one. Alternatively, the cost of a vector operation can be measured as $S + Rp$ where S is a startup time and R is the vector-rate. It will be clear from the discussion that a more detailed analysis for a specific computer can be carried out.

Consider an algorithm where the vectorization is performed on the inner loops of algorithm 4.2 whenever possible, resulting in an algorithm closely related to the sequential method. Assume that $p \leq N$ processors are available. The description references the steps of algorithm 4.2.

Algorithm 4.5.

Steps 1 and 8.

a) The setup time scaling the right hand side, takes N^2/p timesteps using $p \leq N$ processors.

b) The remainder of step 1 and all of step 8 is computed by using a sequential fast Fourier transform algorithm on p independent vectors in parallel. The total time for this is $4N^2 \log N/p$ using $p \leq N$ processors.

Steps 3 and 5.

a) The subalgorithm PENTF (i,r,y,x) as stated in section 4.2, consists of two vector operations forming w, two

vector innerproducts when computing c_1 and c_2, and finally two vector operations when calculating x. All vectors in this subroutine have length $N/2$. Assuming that an innerproduct between two vectors of length N requires $N/p + \log p$ vector operations, the cost of PENTF (i,r,y,x) is:

i) $3N/p + 2\log p$ if no preprocessing is done.

ii) $5N/2p + \log p$ if c_1 is precomputed.

iii) $3N/2p + \log p$ if c_1 and w are precomputed.

Notice that these results are valid for $p \leq N/2$.

b) Therefore, assuming (as in section 4.2) that only c_1 is precomputed, steps 3 and 5 require approximately $N(13N/p + 4\log p)$ timesteps with $p \leq N$.

Step 4.

a) Using the same assumptions, a matrix-vector product takes $\frac{N}{4}(6N/p + 2\log p)$ timesteps using $p \leq N/2$ processors.

b) All other operations in the conjugate gradient iteration, including the preconditioning step, can be performed in $O(N/p) + O(\log p)$ timesteps per iteration. Since only k iterations are needed, k independent of N, the cost of step 4 solving all four linear systems, is approximately $kN(6N/p + 2\log p)$ with $p \leq N$. The total time required for this algorithm is therefore

$$\frac{N^2}{p}(2\log N^2 + 6k + 14) + 2N(k+2)\log p \quad . \qquad (4.13)$$

Notice that the first term in this expression is (4.4) divided by p. For large N and $p << N$ the speedup is very close to p. If $p = N$

the operation count becomes

$$N(2(k+4)\log N + 4k + 10) \quad , \tag{4.14}$$

since the $\log p$ term arising from inner products of vectors of length $N/2$, never exceeds $\log(N/2)$. In this case the speedup is proportional to N/k.

As an illustration, a computer implementation of algorithm 4.2 was tried on an IBM 370/168 and also on the CRAY-1 computer. Figure 4.10 displays some timing results.

	N = 255		N = 511	
	SOLV	FFT	SOLV	FFT
IBM 370/168	7109	4324	____	____
CRAY-1 OFF = v	1804	882	7299	3506
CRAY-1 ON = v	251	548	878	2148

Figure 4.10 Time in milliseconds to solve the biharmonic equation on an N by N grid.

Remarks.

i) The total solution time is the sum of the time spent in a fast Fourier transform routine (FFT) and in the remainder of the code (SOLV).

ii) The FORTRAN H(OPT=3) compiler was used on the 168, while the CFT compiler on the CRAY-1 was used with and without the vectorization option. (ON = v and OFF = v).

iii) The same FORTRAN source code was used in all three cases with the single exception that a special vector inner product routine written by Oscar Buneman [1980], was used in

the vectorized run.

iv) The iterative part of the algorithm was terminated when the 2-norm of the residual fell below the tolerance $TOL = 10^{-10}$.

v) No attempt was made to optimize the code by avoiding nonvectorizable $O(N)$ contributions to the execution time. In particular, the Fourier transform part of the code was not implemented as in algorithm 4.5, and therefore executes slowly.

vi) Notice the substantially improved execution time for the SOLV-part when vectorization is turned on. The speedup compared with scalar processing, is between seven and eight, while the Fourier transform routine only gains a factor 1.6. The algorithm is sufficiently parallel in its structure that a FORTRAN program written for a sequential computer, immediately speeds up when given to the CFT-compiler on the CRAY-1.

An alternative to algorithm 4.5 is to perform all the vectorizations on the outer loops. The resulting code will differ more from a sequential implementation, but it avoids the difficulty with the vector inner-products in algorithm 4.5. The following is a brief description again refering to algorithm 4.2. Steps 1 and 8 will be as in algorithm 4.5. In steps 3 and 5 the vectorization must be performed on the index i, resulting in a cost of $13N^2/p$ timesteps. Similarly in step 4, the cost of a matrix-vector product becomes $3N^2/2p$ timesteps using $p \leq N/2$. The cost of solving all four systems is therefore $6kN^2/p$ resulting in a total cost of

$$\frac{N^2}{p}(2\log N^2 + 6k + 14) \quad . \tag{4.15}$$

Comparing this with (4.4) shows that an optimal speedup of p has been achieved.

Finally, consider the case where a parallel computer with N^2 processors, as described in Sameh, Chen and Kuck [1976], is being used. Using their results on computing the fast Fourier transform, and a combination of the two algorithms outlined in this section, it can be shown that a method based on algorithm 4.2 can be executed in $O(\log N)$ timesteps. This is an order of magnitude faster than results of Sameh, Chen and Kuck and its complexity is the same as that of a Poisson solver under similar assumptions.

Remark.

The main purpose of this section is to outline results when using multiprocessor computers. These results can also be useful when considering algorithms for vector computers. The particular operation counts are of the right order of magnitude, but the constants can certainly be improved by departing in certain respects from the particular underlying sequential algorithm 4.2.

4.6 Roundoff errors.

The numerical algorithms proposed in this Chapter, are all solving a linear system of equations with coefficient matrix A given by (3.2). The condition number of this matrix is proportional to N^4. Classical theory for linear equations (Wilkinson [1965]), shows that there exist a right hand side and a perturbation of this vector, that result in errors in the solution proportional to the condition number times the original

perturbation. A numerical method for this problem is said to be stable when the error due to roundoff is bounded by a constant times the condition number. Several authors (Strang and Fix [1973], Schröder, Trottenberg and Witsch [1978]) have pointed out that in the case of discrete systems derived from certain differential equations, even more stable numerical methods are conceivable.

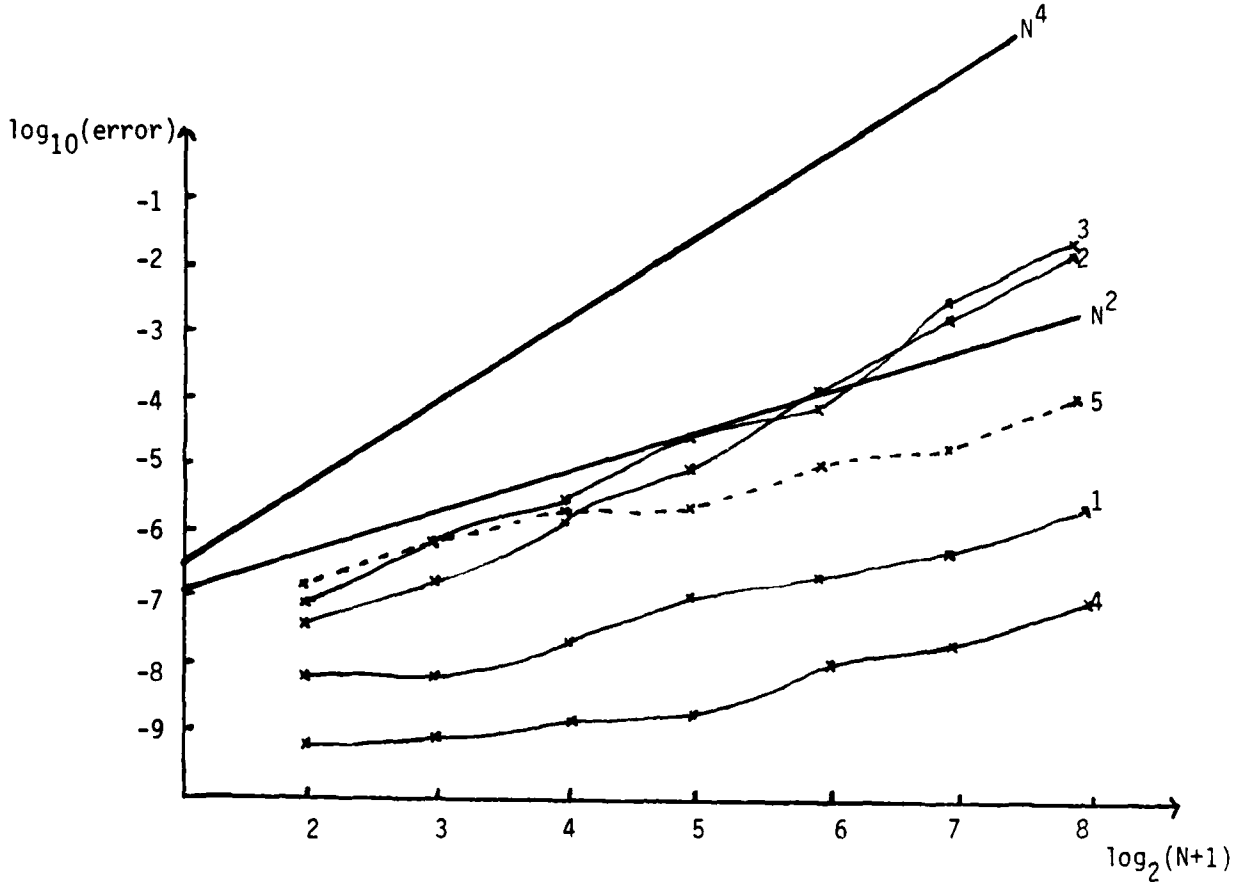


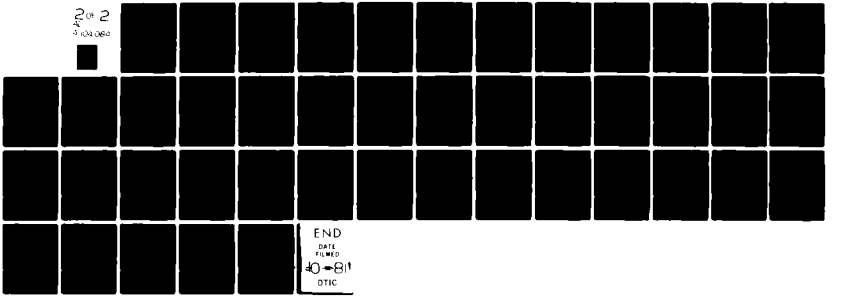
Figure 4.11 Maximum roundoff error for problems 1 through 5 as a function of N.

These questions have not received much attention, mainly because the truncation error usually is more important as long as the numerical method is stable. With the development of fast methods, in particular for fourth order problems, this is no longer necessarily true.

In order to study the roundoff error, algorithm 4.3 was tried in both single and double precision. The difference between the two results were computed for the five first test problems. The maximum roundoff error on the grid, propagating from the inexact representation of the right hand side in single precision is shown in figure 4.11. The computer used was a VAX-11/780 with single precision machine epsilon of $3 \cdot 10^{-8}$.

Lines indicating growth in errors proportional to N^4 and N^2 are indicated as references. The figure shows a difference between highly symmetric problems (1, 4 and 5) and more general problems like 2 and 3. The analysis of this difference will not be pursued here, but it seems clear that it is related to the fact that most of the linear systems in step 4 of algorithm 4.2 are essentially trivial in the symmetric cases. The figure indicates that the roundoff error stays well below the N^4 reference lines in all cases.

AD-A104 084 STANFORD UNIV CA DEPT OF COMPUTER SCIENCE F/G 12/1
NUMERICAL SOLUTION OF THE BIHARMONIC EQUATION,(U)
DEC 80 P E BJORSTAD N00014-75-C-1132
UNCLASSIFIED STAN-CS-80-834 NL



4.7 Efficient solution of the discrete system when the cubic extrapolation scheme is used near the boundary.

The stencil (2.4) employing cubic extrapolation near the boundary, leads to a nonsymmetric linear system of equations with a slightly more complicated structure than (3.2). The two finite difference schemes are of the same order of accuracy when computing the discrete solution u, but as shown in Chapter II, there are cases when the cubic extrapolation procedure is preferable.

It has been claimed (Gupta [1979]), that fast methods for the classical approximation using quadratic extrapolation, cannot be used and that more general, but expensive methods are necessary when the cubic extrapolation is used. The present section outlines two new fast methods based on the algorithm developed in Chapter III.

First, consider the possibility of deriving a numerical method using the same ideas as in Chapter III. Without loss of generality it is assumed that $N = M$. The coefficient matrix A can be written as

$$A = \left[(I \otimes R) + (R \otimes I)\right]^2 + (T_c \otimes I) + (I \otimes T_c) \qquad (4.16)$$

where

$$T_c = 4(e_1 e_1^T + e_N e_N^T) - \tfrac{1}{2}(e_1 e_2^T + e_N e_{N-1}^T) \quad . \qquad (4.17)$$

Let

$$T_c = UV^T$$

where

$$U = \tfrac{1}{2}\begin{pmatrix} 8 & -1 & 0 & 0 \\ & & 0 & \\ 0 & 0 & -1 & 8 \end{pmatrix}_{N\times 4}, \quad V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ & & 0 & \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{N\times 4} .$$

Proceeding as in Chapter III,

$$P(I \otimes Q)A(I \otimes Q)P^T = \hat{S} + (X \otimes I)(Y \otimes I)^T \qquad (4.18)$$

where $X = QU$ and $Y = QV$.

$\hat{S}$ is blockdiagonal and each block has a pentadiagonal slightly nonsymmetric structure. This poses no real difficulty and fast methods for solving linear systems involving $\hat{S}$ can be devised. The matrix B (3.11), now takes the form

$$B = I + (Y \otimes I)^T \hat{S}^{-1}(X \otimes I) \quad . \qquad (4.19)$$

This matrix has a 4×4 blockstructure with blocksize N. By going through the same calculations that led to (3.16) and (3.17), this matrix decouples into two 2×2 block matrices. The systems can be further reduced by one step of block Gaussian elimination. The resulting $N \times N$ matrices can be generated and then factored using the LU - decomposition. Another, perhaps more interesting idea is to solve these systems using an iterative method. As an example, one of the linear systems that must be solved has the following structure

$$(I + \frac{16}{N+1} \sum_{k=1,3,5..}^{N} \sin^2 \frac{k\pi}{N+1} S_k^{-1})x = b \quad , \qquad (4.20)$$

where S_k is the k-th block of $\hat{S}$.

Proceeding as in Chapter III, a good preconditioning matrix will be obtained by replacing S_k^{-1} by $\tilde{S}_k^{-1}$. Since the problem is nonsymmetric, the conjugate gradient method cannot be used in the same way as before. The normal equations can always be used in combination with the conjugate gradient method, but such an approach is unnecessarily expensive when dealing with this problem. Various extensions of the conjugate gradient method have been considered, see Kincaid and Young [1980] and others. Theoretical understanding of these methods is not complete.

Consider the class of quasi-Newton updates proposed by Broyden [1965], for solving systems of nonlinear equations. Given a nonsingular $n \times n$ matrix H_0, an initial guess x_0 and $r_0 = Ax_0 - b$, for $k = 0,1,2...$ let

$$\begin{aligned} s_k &= -H_k r_k \ , \\ x_{k+1} &= x_k + s_k \ , \\ y_k &= As_k \ , \\ r_{k+1} &= r_k + y_k \ , \\ H_{k+1} &= H_k + (s_k - H_k y_k) v_k^T \ , \end{aligned} \tag{4.21}$$

where $v_k^T y_k = 1$ and $v_k^T H_k^{-1} s_k \neq 0$.

Gay [1979] has shown that this process converges in at most $2n$ steps to the solution of the linear system. Moreover, computational experience indicates that convergence often is very rapid when the spectrum of AH_0 is clustered. The symmetric rank one update discussed in section 4.4, is obtained by setting $v = (s-H_y)/y^T(s-H_y)$ and this method was tried when solving nonsymmetric systems like (4.20). When using the proper preconditioning matrix H_0, convergence is very rapid. This is plausible since the algorithm converges in at most $2p$ steps if AH_0 has only p distinct eigenvalues.

A limited storage version of (4.21) can be implemented requiring only the two matrix-vector products $y = As$ and $v = Hy$ in addition to a few vector operations per iteration. The required storage is only three vectors in addition to K ($K \geq 1$) vectors for the updates. This also suggests an alternative way of solving the original discrete problem by applying algorithm (4.21) directly, defining H_0 by one of the fast algorithms discussed in the beginning of this Chapter. This technique was used when producing the results in Chapter II. Figure 4.12 shows the number of iterations required to solve problems 1 and 2 on four different grids.

Only 5 updates and a total storage requirement of $8N^2$ was used. The iteration was stopped when $||Ax-b||_2 < (N+1)TOL$.

N	15	31	63	127
Problem 1, TOL=10^{-6}	5	4	4	3
Problem 1, TOL=10^{-12}	10	9	9	7
Problem 2, TOL=10^{-6}	9	9	9	8
Problem 2, TOL=10^{-12}	15	15	15	15

Figure 4.12 Number of iterations required when using Broyden's method.

After 5 iterations the method was restarted with $H_6 \equiv H_0$. Experiments show that this is more efficient than using $H_{k+1} \equiv H_k$ for $k \geq 5$. Notice that the number of iterations tend to decrease with increasing N. The required work to solve this problem is therefore of the same order as in the case when a quadratic boundary extrapolation is used.

Remarks.

i) Both H_0^{-1} and A are discrete approximations to the biharmonic operator, but the approximations are different and not

necessarily very accurate ((2.3) and (2.4)) near the boundary.

ii) Broyden's method used in this manner promises to be advantageous in other similar situations as well.

iii) The use of Broyden's method to solve large nonsymmetric systems of linear equations appears to be new. The symmetric rank one update is of particular interest in this context, since it also belongs to the Broyden β-class and therefore is related to the conjugate gradient method in the symmetric case. The method behaves very similar to the conjugate gradient method, but can be used to handle a larger class of problems.

The symmetric rank one update has some theoretical difficulties since $s-Hy$ and y can become orthogonal. This never caused any problems in the applications tried here, but a study of theoretical aspects using one of Broydens single rank updates, when solving large linear systems is planned for the near future.

4.8 Efficient solution of the biharmonic equation in a disk.

Consider the biharmonic Dirichlet problem on a disk of radius R,

$$\begin{aligned} \Delta^2 u &= f \quad r < R \\ u &= g \quad r = R \\ u_r &= h \quad r = R \ . \end{aligned} \tag{4.22}$$

In polar coordinates the biharmonic operator takes the form

$$\Delta^2_{r,\theta} = \frac{1}{r}(\partial_r(r\partial_r) + \frac{1}{r}\partial^2_\theta)\,\frac{1}{r}(\partial_r(r\partial_r) + \frac{1}{r}\partial^2_\theta) \ . \tag{4.23}$$

Glowinski and Pironneau [1979] remarked that a discrete form of this

problem derived from a finite difference grid based on polar coordinates, can be solved by using the "coupled equation approach". (See Chapter I, section viii).) The present section describes an algorithm which is an order of magnitude faster. Taking advantage of the explicit formula (1.8) valid when $f \equiv 0$, makes it possible to design a direct method for (4.22). Let

$$u = u_1 + u_2 \quad .$$

First solve

$$\begin{aligned} \Delta w_1 &= f & r < R \\ w_1 &= 0 & r = R \end{aligned} \tag{4.24}$$

and then

$$\begin{aligned} \Delta u_1 &= w_1 & r < R \\ u_1 &= g & r = R \end{aligned} \tag{4.25}$$

The problem for u_2 becomes

$$\begin{aligned} \Delta^2 u_2 &= 0 & r < R \\ u_2 &= 0 & r = R \\ (u_2)_r &= h - (u_1)_r & r = R \end{aligned} \tag{4.26}$$

Now write

$$u_2 = -(R^2 - r^2)v_1 + v_2$$

and require that v_1 and v_2 be harmonic (see 1.7). Since v_2 vanishes at the boundary, it follows that it is identically zero. Now

$$\left(\frac{\partial u_2}{\partial r}\right)_{r=R} = 2Rv_1 \quad ,$$

and therefore

$$\begin{aligned} \Delta v_1 &= 0 & r < R \\ v_1 &= \frac{1}{2R}(u_2)_r & r = R \quad . \end{aligned} \tag{4.27}$$

In this way the numerical solution of (4.22) has been reduced to the solution of three Poisson equations on the same grid. The derivative $(u_1)_r$ which is needed in (4.26) must be computed with sufficient accuracy from the solution of (4.25). If a second order method is used for solving Poisson's equation then the discrete value of $(u_1)_r$ should also be second order accurate. A second order accurate numerical solution to the original (smooth) problem (4.22) can then be obtained.

A computer implementation using the subroutine PWSPLR (Swarztrauber and Sweet [1975]) for Poisson's equation, has been written. The algorithm has an operation count of $O(NM\log N)$ when a discretization with N points in the θ-direction and M points in the r - direction is used. A somewhat faster code requiring less storage, could be implemented by taking advantage of the zero right hand side in (4.27). Figure 4.13 displays the results from a test using an IBM 370/168 computer. The problem was solved in the unit disk and the exact solution is given by $u = e^{r\sin\theta}$.

M	N	TIME (MS)	MAX ERROR
32	32	483	$4.89\ 10^{-4}$
64	64	2158	$1.22\ 10^{-4}$
128	128	9670	$3.05\ 10^{-5}$

Figure 4.13 Execution time and discretization error when solving the biharmonic equation in a disk.

It is easily seen that the discrete solution is second order accurate.

4.9 Conformal mapping and the solution of the biharmonic equation on more general domains.

Given a domain $\Omega_w \subset R^2$, with boundary $\partial\Omega_w$, consider the biharmonic Dirichlet problem,

$$\begin{aligned} \Delta_w^2 u &= f \quad \text{in} \quad \Omega_w \\ u &= g \quad \text{on} \quad \partial\Omega_w \\ u_n &= h \quad \text{on} \quad \partial\Omega_w \quad . \end{aligned} \tag{4.28}$$

Assume that it is possible to map the unit disk conformally to Ω_w, i.e. the mapping function or a sufficiently accurate approximation is known or efficiently computable. This is indeed the case for all polygons, since the map in this case called the Schwarz-Christoffel transformation, has a simple form and can be accurately computed. (Trefethen [1980]). There are also methods for computing the map to more general domains, see for example Gaier [1964], Symm [1966], Chakravarthy and Anderson [1979], Gutknecht [1980], Fornberg [1980]. This section outlines how it is possible to solve (4.28) taking advantage of fast solution techniques developed for a rectangular region. (The conformal map between a rectangle and the disk is an easy problem). The advantage of this approach is having a fixed computational domain where highly specialized numerical methods can be used. However, the necessary calculation of the map can often be difficult and dominate the computational cost. In some applications this can be considered as preprocessing if many problems of the form (4.28) are solved for a fixed domain.

Assume in what follows that $s(z)$ maps the given rectangle conformally to the domain Ω_w. For any function $f(w)$ $w \in \Omega_w$, let $f^{(s)}(z)$ denote the function such that $f^{(s)}(z) \equiv f(s(z))$. First write equation

(4.28) as two coupled second order equations,

$$\begin{aligned} -\Delta_w u &= v \quad \text{in} \quad \Omega_w \\ -\Delta_w v &= f \quad \text{in} \quad \Omega_w \\ u &= g \quad \text{on} \quad \partial\Omega_w \\ u_n &= h \quad \text{on} \quad \partial\Omega_w \quad . \end{aligned} \tag{4.29}$$

The equivalent problem in the computational domain is

$$\begin{aligned} -\Delta_z u^{(s)} &= |s'(z)|^2 v^{(s)} \quad \text{in} \quad R_z \\ -\Delta_z v^{(s)} &= |s'(z)|^2 f^{(s)} \quad \text{in} \quad R_z \\ u^{(s)} &= g^{(s)} \quad \text{on} \quad \partial R_z \\ u_n^{(s)} &= |s'(z)| h^{(s)} \quad \text{on} \quad \partial R_z \quad . \end{aligned} \tag{4.30}$$

This problem can be discretized using the standard stencils discussed in Chapter II. Let S be the positive diagonal matrix containing the discrete values of $|s'(z)|$ on the gridpoints. If the quadratic extrapolation scheme is used near the boundary the discrete matrix problem representing (4.30) is

$$(L\, S^{-2}\, L + UU^T) u_h^{(s)} = h^4\, S^2\, f_h^{(s)} + \ell \tag{4.31}$$

where L is the discrete Laplacian, U is an $N^2 \times 4(N-1)$ matrix and ℓ is a sparse vector. Notice that this problem has the same structure as the problem discussed in Chapter III except that the diagonal matrix S has been introduced. The $4(N-1) \times 4(N-1)$ matrix

$$B = (I + U^T L^{-1} S^2 L^{-1} U) \tag{4.32}$$

can be generated and factored in $O(N^3)$ operations. The linear system

can also be solved using conjugate gradients. The preconditioning technique employed in Chapter III cannot be used in this case. There remains to investigate possible alternatives. When $S = I$ the eigenvalues of B approximately equals cN/i for $i = 1,2,3..$. The conjugate gradient method requires $O(N^{7/3})$ arithmetic operations to solve (4.31) for a problem with such a spectrum.

Finally it should be mentioned that this technique can be used in combination with a Lanczos eigenvalue routine when solving the eigenvalue problem associated with (4.28).

CHAPTER V

APPLICATIONS

The existence of an efficient numerical method for solving the generalized biharmonic equation (4.1) makes it a useful computational tool in the construction of numerical methods for more complicated fourth order problems much in the same way as fast Poisson solvers have been used in the past ten years.

Problems where this numerical method may prove useful include von Kármán's equation (1.5) and the streamfunction formulation of Navier Stokes equation for incompressible flow (Temam [1977]). A class of problems closely related to (4.1), including physical examples, is discussed by A. and M.B. Banerjee, Roy and Gupta [1978]. In some of these applications a nonuniform (graded) mesh may be advantageous. Extension of the numerical method to more general fourth order equations having separable lower order terms, is not difficult and this makes it possible to handle certain coordinate transformations introducing such meshes.

Two applications using the numerical methods for equation (4.1) will be briefly discussed in the remainder of this Chapter.

5.1 The eigenvalue problem for the biharmonic operator.

Consider the eigenvalue problem

$$\begin{aligned} \Delta^2 u &= \lambda^2 u \quad \text{in } R \\ u &= 0 \quad \text{on } \partial R \\ u_n &= 0 \quad \text{on } \partial R \ , \end{aligned} \tag{5.1}$$

in a rectangle R. This problem defines the natural frequencies and the natural modes of vibration of a clamped elastic plate. Formulation (4.1)

can be used with $\alpha = 0$, in a Rayleigh quotient iteration when computing approximations to the lowest modes. Due to the cubic rate of convergence, this method is more efficient than some of the previous methods that have been tried. (Bauer and Reiss [1972]). A modified form of algorithm 4.3 was used when solving the resulting sequence of indefinite problems. The Fourier transforms in the algorithm can be omitted when doing this iteration. After the transformed eigenvectors have converged, they may be Fourier transformed in a postprocessing stage resulting in a substantial savings in computational work.

This section briefly describes the relationship between the numerical method and the space of eigenfunctions. In addition, the behavior of the first eigenfunction near a corner is studied.

A particular eigenfunction is generated by only one (or in the degenerate case by two) of the matrices C^{rs} given in theorem 3.1. The notation (r,s) $r,s = 1,2$, will be used to indicate which of the four problems in step 4 of algorithm 4.3 that must be solved for a given eigenvalue. This results in additional computational savings and also provides a more systematic way of studying the eigenfunctions. Figure 5.1 lists the five first distinct eigenvalues obtained by extrapolating from solutions using $N = 63$ and $N = 127$. More accurate calculations can easily be performed by going to finer grids or even better, by using the matrices given in lemma A2.2. Good upper and lower bounds have been published by Fichera [1966] and they are included in figure 5.1.

	Lower bound	Estimate	Upper bound
λ_1	35.9852	35.9852	35.9852
$\lambda_{2,3}$	78.3922	73.3937	73.3939
λ_4	108.213	108.216	108.217
λ_5	131.573	131.580	131.581
λ_6	132.197	132.220	132.220

Figure 5.1 The first five distinct eigenvalues computed using N=63 and N=127 compared with lower and upper bounds.

In these calculations R was taken to be the unit square and λ is defined by (5.1). In order to relate the decomposition of the eigenspace to the symmetries of the eigenfunctions and the previous work of Fichera, let (x,y) be a point in the first quadrangle of the square with $x > y$. Consider the eight points $p_i = \{(x,y), (y,x), (-y,x), (-x,y), (-x,-y), (-y,-x), (y,-x)\}$, $i = 1,2,...8$ defining the possible symmetries of a solution defined on R. The following relationships hold:

i) The eigenfunctions with total symmetry, $u(p_i) = u(p_1)$ for $i = 2,3..8$, are generated by C^{11}, this group corresponds to (0000) in Fichera's notation.

ii) The eigenfunctions symmetric around the coordinates axis, but antisymmetric around the diagonals, $u(p_i) = u(p_1)$ $i = 4,5$ and 8, $u(p_i) = -u(p_1)$, $i = 2,3,6$ and 7, are also generated by C^{11}, this group corresponds to (0011) in Fichera's notation.

iii) The eigenfunctions which are antisymmetric under a rotation of π, $u(p_{i+4}) = -u(p_i)$, $i = 1,2,3$ and 4, are generated by the two matrices C^{12} and C^{21}. This is a degenerate case and for each eigenvalue in this group there are two eigenfunctions. The two eigenfunctions have the same shape, but one is rotated $\pi/2$ compared to the other. Fichera calls this case (01-10).

iv) A total antisymmetric eigenfunction, $u(p_{i+1}) = -u(p_i)$, $i = 1,2,3...8$ is generated by C^{22}, corresponding to (1111) in Fichera's notation.

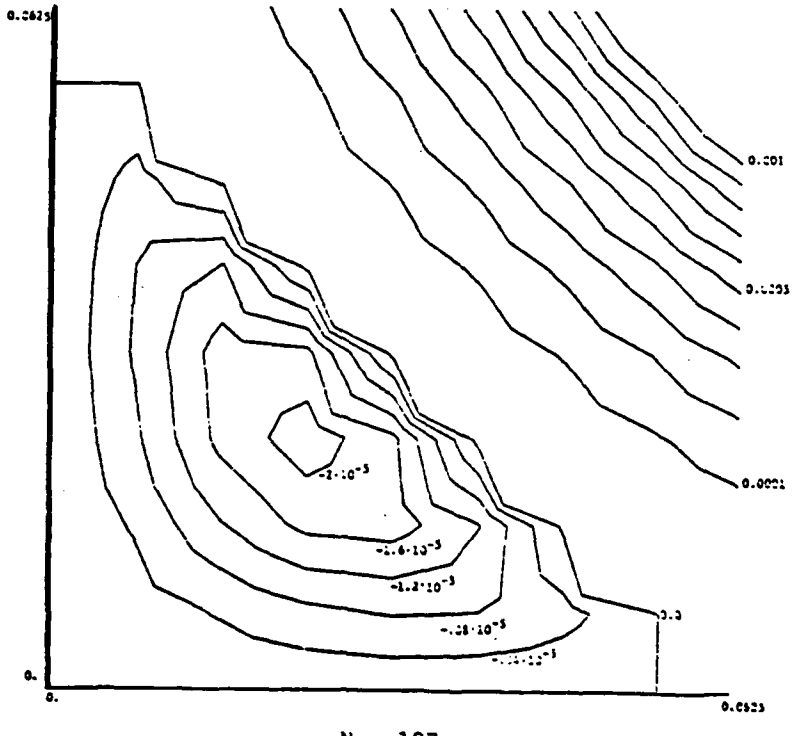
v) An eigenfunction symmetric around the diagonals, but antisymmetric around the coordinate axis, $u(p_i) = u(p_1)$, $i = 2,5$ and 6, $u(p_i) = -u(p_1)$, $i = 3,4,7$ and 8, is also generated by C^{22}, this group is called (1100) by Fichera.

These results are important in order to understand which of the linear systems in step 4 of algorithm 4.2 that are nontrivial for a problem with a given symmetry. (See Appendix III.) The results also indicate that it may be possible to refine the decomposition given in theorem 3.1, by further splitting the matrices C^{rs}.

Next, consider the shape of the first eigenfunction in the neighborhood of a corner. Bauer and Reiss [1972] reported the existence of nodal lines in the vicinity of corners, but their numerical method severely limited a detailed study. Other researchers noticed that the nodal line moved towards the corner as the grid was refined, and questioned its existence in the limit. Theoretically this had been an open question for quite some time. (Very recently, after this investigation was completed,

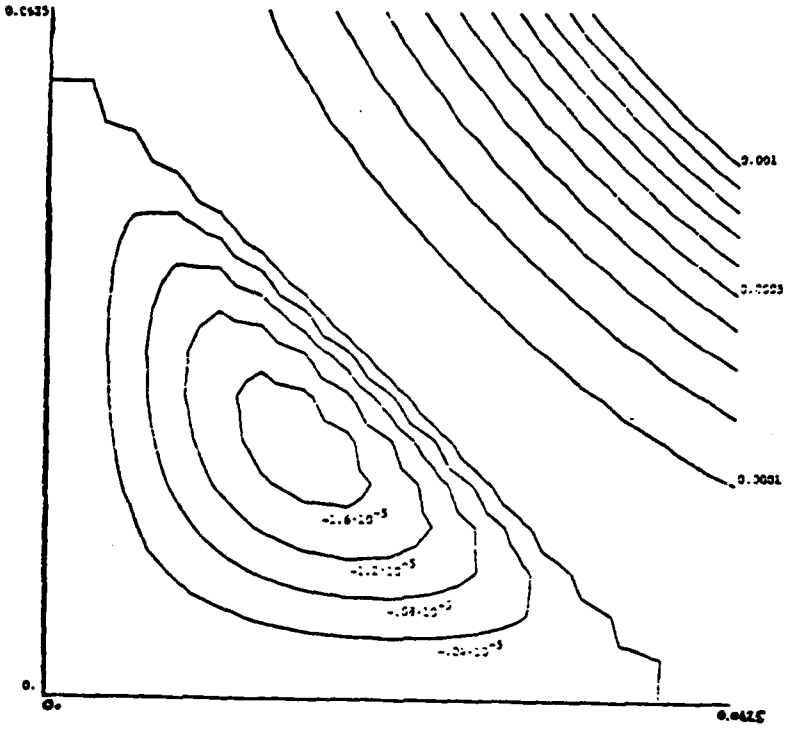
Coffman [1980] informed the author that he had proved the existence of nodal lines.)

The fine grids permitted by the new numerical method made it possible to study this question numerically. The theory in Chapter III and Appendix II may also be used to investigate this phenomenon in the continuous case. Figures 5.2 and 5.3 show contour plots of the first eigenfunction near a corner of the unit square based on calculations using N=127 and N=255. Figure 5.4 shows a surface plot of the same area based on the finest grid. Finally, after normalizing the eigenfunction such that its maximum value is 1, the extrapolated values based on the two grids, are shown in figure 5.5.



N = 127

Figure 5.2 Contour plots of the first biharmonic eigenfunction near a corner.



N = 255

Figure 5.3 Contour plots of the first biharmonic eigenfunction near a corner.

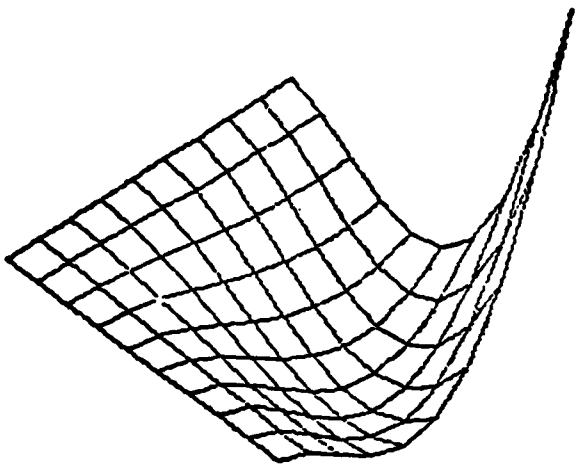


Figure 5.4 Surface plot of the first biharmonic eigenfunction near a corner.

5/128	-4.74	-6.56	+8.64	+50.37	+124.71
4/128	-5.78	-13.09	-10.65	+8.94	+50.37
3/128	-5.12	-13.40	-16.96	-10.65	+8.64
2/128	-3.21	-9.16	-13.40	-13.09	-6.56
1/128	-1.00	-3.21	-5.12	-5.78	-4.74
	x=1/128	x=2/128	x=3/128	x=4/128	x=5/128

y ↑ , x →

Figure 5.5 Extrapolated values of the eigenfunction near a corner. The numbers are scaled up by a factor 10^6.

5.2 Navier Stokes equation.

As an illustration of a problem where a numerical method for (4.1) with nonzero parameter α can be used, consider the driven cavity model problem for the nonlinear, time dependent Navier Stokes equation. Introducing a stream function Ψ in the usual way, the equation was solved using the following scheme:

$$\Delta^2\Psi_{k+1} - \frac{R}{\Delta t}\Delta\Psi_{k+1} = R(\Psi_y\Delta\Psi_x - \Psi_x\Delta\Psi_y)_k - \frac{R}{\Delta t}\Delta\Psi_k \ . \qquad (5.2)$$

Here k denotes the current time level and Δt the time discretization step. The equation was discretized in space using second order accurate centered differences and the 13-point approximation with quadratic boundary extrapolation was used to approximate the biharmonic operator. Notice that this is a special case of (4.1) with nonzero α and $\beta = 0$. The problem was solved in a square region with Reynold's number $R = 200$ and boundary conditions $\Psi = 0$ and $\Psi_n = 0$ except at the side $y = 1$

where

$$\psi_n = \begin{cases} -\sin t & 0 \leq t \leq \pi/2 \\ -1 & \pi/2 \leq t \leq 5 \end{cases} .$$

This corresponds to an acceleration of the moving wall up to the standard velocity used in stationary calculations. A $31 * 31$ grid was used and 500 timesteps each of length 0.01 was taken. (This is smaller than required for stability with this Reynold's number.) The execution time for this problem was approximately one minute on an IBM 370/168. The velocity fields are shown in figures 5.6 and 5.7 at two different times.

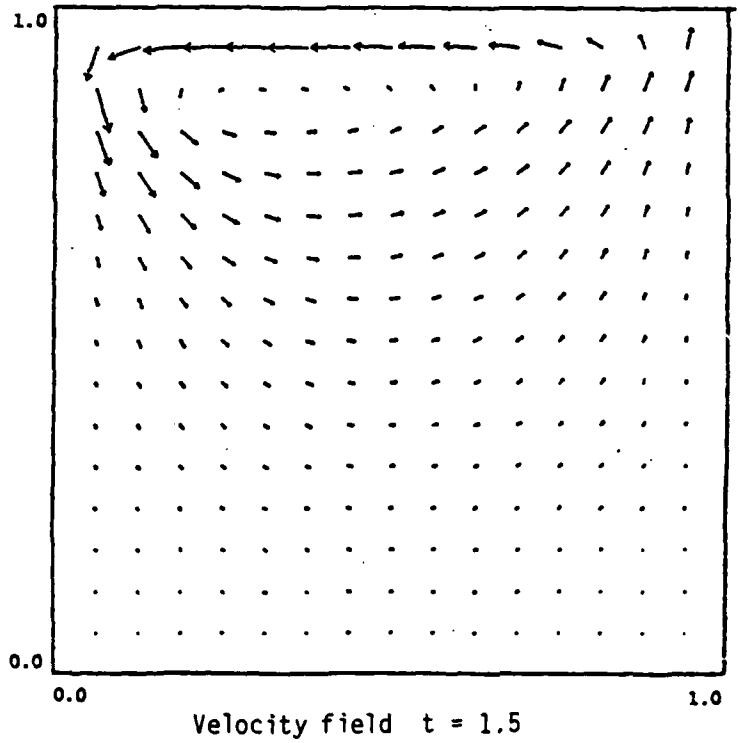
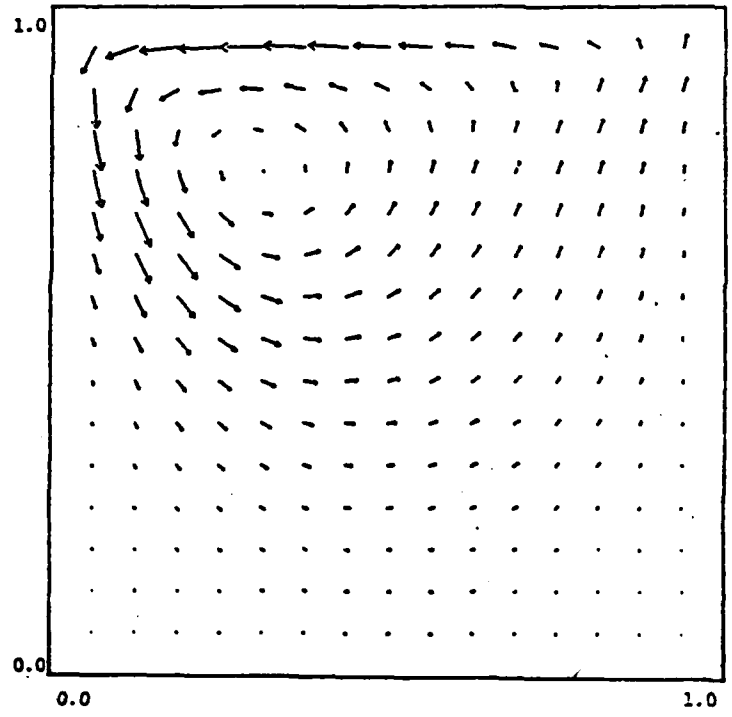


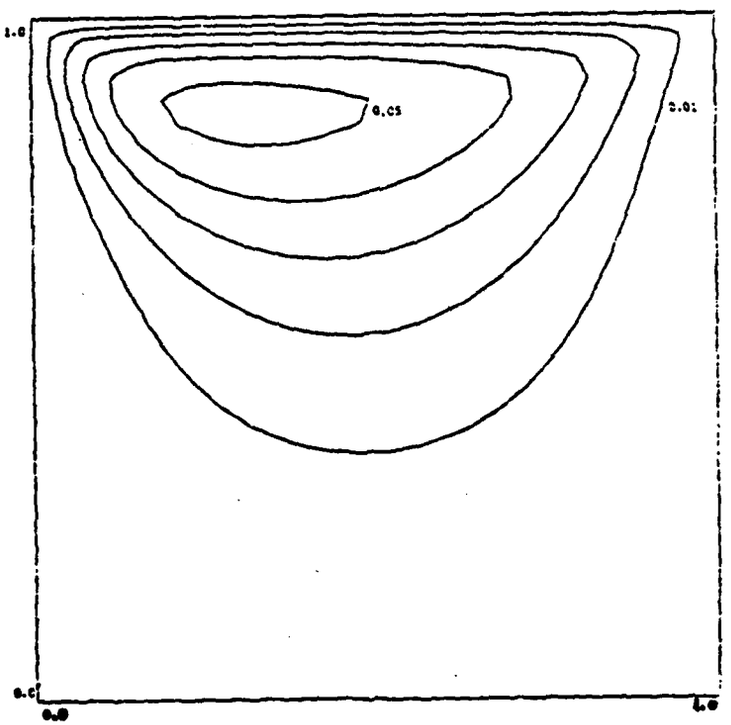
Figure 5.6 Discrete solution of the time dependent Navier Stokes equation at Reynold's number 200.

The corresponding streamfunctions are contour plotted in figures 5.8 and 5.9. The flow is not stationary at time 5, but changes very slowly into a final state after a time equal 20 with a main vortex center $\Psi = 0.105$ at coordinates $x = 0.41$, $y = 0.66$ in good agreement with stationary calculations with this grid.



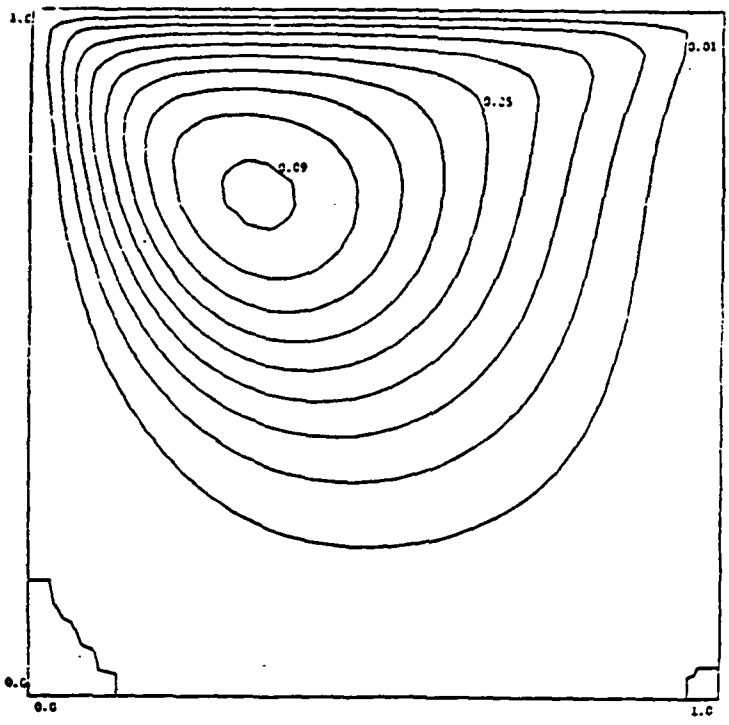
Velocity field t = 5.0

Figure 5.7 Discrete solution of the time dependent Navier Stokes equation at Reynold's number 200.



Stream function t = 1.5

Figure 5.8 Discrete solution of the time dependent Navier Stokes equation at Reynold's number 200.



Stream function t = 5.0

Figure 5.9 Discrete solution of the time dependent Navier Stokes equation at Reynold's number 200.

It should be mentioned that the difference scheme (5.2) is unsatisfactory for large Reynold's number, due to the fact that the nonlinear term is handled in a fully explicit way. Computational experience indicates that the numerical methods developed in this thesis, for the biharmonic equation, can be used as a part of more sophisticated methods when solving the stationary driven cavity problem at large Reynold's number.

APPENDIX I

Proof of Theorem 3.1

The notation used is consistent with the notation in Chapter III. The development in this section is very similar to the derivation of (3.16) and (3.17), but individual components are defined instead of submatrices. An explicit representation for the quantity $Q_N T_i Q_N$ in Theorem 3.1 is needed.

$$Q_N T_i Q_N = I_N + \beta_M \sum_{k=i,i+2,..}^{M} \sin^2 \frac{k\pi}{M+1} Q_N S_k^{-1} Q_N \quad i=1,2 .$$

Let $A_k = Q_N S_k^{-1} Q_N$

$$= \Psi_k^{-1}(I_N - K_N(I_2 + 2 K_N^T \Psi_k^{-1} K_N)^{-1} K_N^T \Psi_k^{-1})$$

where $K_N = Q_N U_N$.

Define the 2×2 matrix

$$B_k = I_2 + 2 K_N^T \Psi_k^{-1} K_N$$

$$= I_2 + \begin{bmatrix} a_k+b_k & a_k-b_k \\ a_k-b_k & a_k+b_k \end{bmatrix}$$

where

$$a_k = \tfrac{1}{2} \beta_N \sum_{j=1,3,5,..}^{N} \sin^2 \frac{j\pi}{N+1} \Psi_{kj}^{-1}$$

and

$$b_k = \tfrac{1}{2} \beta_N \sum_{j=2,4,6,..}^{N} \sin^2 \frac{j\pi}{N+1} \Psi_{kj}^{-1} .$$

It is clear that any 2×2 linear system

$$B_k z = r$$

can be solved in exactly the same way as described in Chapter III for the block case. Recalling the definition of α_k^{rs} in Theorem 3.1, this leads to

$$z_1 + z_2 = (r_1+r_2)/\alpha_k^{1N}$$

$$z_1 - z_2 = (r_1-r_2)/\alpha_k^{2N} \quad .$$

Let e_j be the j'th unit vector of dimension N and consider the j'th column a_j^k of A_k. Observe that

$$K_N^T \Psi_k^{-1} e_j = \sqrt{\frac{2}{N+1}} \sin \frac{j\pi}{N+1} \Psi_{kj}^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{if j is odd}$$

$$= \sqrt{\frac{2}{N+1}} \sin \frac{j\pi}{N+1} \Psi_{kj}^{-1} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \text{if j is even .}$$

Also

$$K_N z = \sqrt{\frac{2}{N+1}} \begin{bmatrix} \sin \frac{\pi}{N+1} & & \bigcirc \\ & \ddots & \\ & & \ddots \\ \bigcirc & & \sin \frac{N\pi}{N+1} \end{bmatrix} \begin{bmatrix} z_1+z_2 \\ z_1-z_2 \\ z_1+z_2 \\ \vdots \\ \vdots \end{bmatrix}$$

Using the above expressions in the definition of A_k gives the following expressions for element a_{ij}^k:

$a_{ij}^k = \Psi_{kj}^{-1} \delta_{ij} - \beta_N \sin \frac{i\pi}{N+1} \sin \frac{j\pi}{N+1} / (\alpha_k^{1N} \Psi_{ki} \Psi_{kj})$ — i,j both odd or i,j both even.

$a_{ij}^k = 0$. — i odd, j even, or i even, j odd.

Therefore

$$P_N A_k P_N^T = P_N \Psi_k^{-1} P_N^T - \beta_N \begin{bmatrix} F_k^1 & 0 \\ 0 & F_k^2 \end{bmatrix}$$

where F_k^r has elements of the form

$$(F_k^r)_{ij} = \frac{\sin \frac{i_r \pi}{N+1} \sin \frac{j_r \pi}{N+1}}{\alpha_k^{rN} \psi_{ki_r} \psi_{kj_r}} \qquad r = 1,2 \quad .$$

This finally gives

$$P_N D_i^{-\frac{1}{2}} Q_N T_i Q_N D_i^{-\frac{1}{2}} P_N^T =$$

$$I - \beta_N \beta_M P_N D_i^{-\frac{1}{2}} P_N^T \sum_{k=i,i+2,..}^{M} \sin^2 \frac{k\pi}{M+1} \begin{bmatrix} F_k^1 & 0 \\ 0 & F_k^2 \end{bmatrix} P_N D_i^{-\frac{1}{2}} P_N^T \qquad i=1,2.$$

Comparing this expression componentwise with the matrix

$$I - (C^{ir})^T C^{ir} \qquad i = 1,2 \qquad r = 1,2$$

in Theorem 3.1 concludes the proof. □

APPENDIX II

Proof of Theorem 3.2 and Theorem 3.3.

Lemma A2.1

$$\text{If } S_N = \sum_{j=1}^{N} \frac{\sin^2 \frac{j\pi}{N+1}}{(B-\cos \frac{j\pi}{N+1})^2} \qquad B > 1$$

then

$$S_N = 2(N+1)\left\{\frac{a^2}{1-a^2} - \frac{(a^{N+1})^2}{1-(a^{N+1})^2}\left[\frac{2(N+1)}{1-(a^{N+1})^2} - \frac{1+a^2}{1-a^2}\right]\right\}$$

where $a = B - \sqrt{B^2-1}$. $\qquad 0 < a < 1 \qquad B - \sqrt{B^2-1}$

Proof:

Define $f(x) = 4\, a^2 \dfrac{\sin^2 x}{(1+a^2-2a \cos x)^2}$.

Fettis [1979] pointed out that the application of Poisson's summation formula to this function gives the relation

$$\tfrac{1}{2}f(0) + f(\tfrac{\pi}{N+1}) + f(\tfrac{2\pi}{N+1}) + \ldots \tfrac{1}{2} f(\pi) = \frac{N+1}{\pi}\left[F_0 + 2 \sum_{k=1}^{\infty} F_{2k(N+1)}\right]$$

where

$$F_m = \int_0^{\pi} f(x) \cos mx \, dx$$

is a cosine transform of $f(x)$ (Magnus and Oberhettinger [1948, p. 217]. Integration by parts yields

$$F_m = 4\, a^2 \int_0^{\pi} \frac{\sin x}{(1+a^2-2a \cos x)^2} \cdot \sin x \cos mx \, dx$$

$$= 2\, a \int_0^{\pi} \frac{\cos x \cos mx}{(1+a^2-2a \cos x)} dx - 2\, a\, m \int_0^{\pi} \frac{\sin x \sin mx}{(1+a^2-2a \cos x)} dx \, .$$

These integrals are well known and can be found in Gradshteyn and Ryzhik [1965, p. 366-367].

$$\int_0^{\pi} \frac{\cos x}{(1+a^2-2a\cos x)}\,dx = \frac{\pi a}{1-a^2} \qquad a^2 < 1 \quad m=0$$

$$\int_0^{\pi} \frac{\cos x \cos mx}{(1+a^2-2a\cos x)}\,dx = \frac{\pi}{2} a^{m-1} \frac{1+a^2}{1-a^2} \qquad a^2 < 1 \quad m=1,2,3$$

$$\int_0^{\pi} \frac{\sin x \sin mx}{(1+a^2-2a\cos x)}\,dx = \frac{\pi}{2} a^{m-1} \qquad a^2 < 1 \quad m=1,2,3 \ .$$

Therefore

$$F_0 = \frac{2\pi a^2}{1-a^2}$$

$$F_m = \pi a^m\left[\frac{1+a^2}{1-a^2} - m\right] \qquad m=1,2,3\ldots \ ,$$

and

$$S_N = 2(N+1)\frac{a^2}{1-a^2}\left[1 + \frac{1+a^2}{a^2}\sum_{k=1}^{\infty}(a^{2(N+1)})^k - 2(N+1)\frac{1-a^2}{a^2}\sum_{k=1}^{\infty}k(a^{2(N+1)})^k\right]$$

$$= 2(N+1)\left[\frac{a^2}{1-a^2} - \frac{(a^{N+1})^2}{1-(a^{N+1})^2}\left(\frac{2(N+1)}{1-(a^{N+1})^2} - \frac{1+a^2}{1-a^2}\right)\right] \ . \ \square$$

Remark:

$$S_N = 2(N+1)\frac{a^2}{1-a^2} + O(a^{2(N+1)})$$

$$= (N+1)\left(\frac{B}{\sqrt{B^2-1}} - 1\right) + O(a^{2(N+1)}) \ .$$

This is the approximation obtained if S_N is approximated by

$$S_N \approx \frac{N+1}{\pi} \int_0^{\pi} f(x)dx \quad .$$

All the error terms in the Euler-McLaurin formula (Dahlquist and Björck [1974, p. 297]) are zero in this case, since $f^{(k)}(0) = f^{(k)}(\pi) = 0$ for k odd. This much simpler expression is always an upper bound for S_N since

$$\frac{2(N+1)}{1-a^{2(N+1)}} - \frac{1+a^2}{1-a^2} \geq 0 \quad .$$

Now consider

$$S_{even} = \sum_{j=2,4,6}^{N} \frac{\sin^2 \frac{j\pi}{N+1}}{(B - \cos \frac{j\pi}{N+1})^2} \qquad B > 1 \ , \ N \text{ odd} \ .$$

Claim:

$$S_{even} = (N+1)\left\{\frac{a^2}{1-a^2} - \frac{a^{N+1}}{1-a^{N+1}}\left[\frac{N+1}{1-a^{N+1}} - \frac{1+a^2}{1-a^2}\right]\right\} .$$

Proof:

Replace $N+1$ by $\frac{N+1}{2}$ in the proof of Lemma A2.1. It is easily seen that the proof still holds. □

The sum over only odd j can now be found as the difference between S_N and S_{even}.

The above derivation furnishes a closed form expression for the quantity α_k^{iN} defined in Theorem 3.1 and therefore closed form expressions for the individual matrix elements c_{ij}^{rs} also given in Theorem 3.1.

An upper bound for the largest singular value σ_1 of the matrices C^{rs} will be derived. The following well known inequality will be used: $\sigma_1 \leq (\| (C^{rs})^T C^{rs} \|_\infty^{\frac{1}{2}} \leq (\| C^{rs} \|_1 \| C^{rs} \|_\infty)^{\frac{1}{2}} = \left[(\max_j \sum_i c_{ij}^{rs})(\max_i \sum_j c_{ij}^{rs}) \right]^{\frac{1}{2}}$ since all matrix elements are positive.

For fixed i and j an element c_{ij}^{rs} increases with the dimension N of the matrix. The interesting case to consider is the limiting behavior as N becomes large. The following important Lemma gives the precise form of the limit matrix.

Lemma A2.2 The matrix C^{rs} defined in Theorem 3.1 has elements:

$$c_{ij}^{rs} = \frac{8}{\pi} \frac{i_r j_s \sqrt{i_r j_s}}{(i_r^2 + j_s^2)^2} \cdot \frac{a_i^{rs} a_j^{sr}}{b_i^{rs} b_j^{sr}} + O\left(\frac{1}{(N+1)^2}\right) \qquad \begin{matrix} r=1,2 \\ s=1,2 \end{matrix}$$

where a_j^{rs} and b_j^{rs} are exponentially close to 1 in j and given by

$$a_j^{rs} = (1+(-1)^{k_{rs}} e^{-j_r \pi})$$

$$b_j^{rs} = (1+2(-1)^{k_{rs}} j_r \pi e^{-j_r \pi} - e^{-2j_r \pi})^{\frac{1}{2}}$$

and

$$k_{rs} = 0 \quad \text{if} \quad r = s = 1 \quad \text{or} \quad r = 2,\ s = 1$$

$$k_{rs} = 1 \quad \text{if} \quad r = s = 2 \quad \text{or} \quad r = 1,\ s = 2$$

$$j_r = 2j - 1 \quad \text{if} \quad r = 1$$

$$j_r = 2j \quad \text{if} \quad r = 2 \ .$$

Proof:

Derive Taylor expansions for each element c_{ij}^{rs} in the variable $\frac{1}{N+1}$ around zero. This is rather tedious to do by hand and the symbolic manipulation program MACSYMA [1977] was used when deriving the above expressions. □

The 3 by 3 leading principal minors of the (infinite) limit matrix C_{∞}^{rs} are compared with the corresponding minors of C_{63}^{rs} for N=63 in Figure A2.1. It is interesting to observe that the approximation is quite good already for this value of N.

$$C_{63}^{11} = \begin{vmatrix} .545 & .122 & .038 \\ .122 & .209 & .125 \\ .038 & .125 & .123 \end{vmatrix} \qquad C_{\infty}^{11} = \begin{vmatrix} .546 & .122 & .039 \\ .122 & .212 & .128 \\ .039 & .128 & .127 \end{vmatrix}$$

$$C_{63}^{12} = \begin{vmatrix} .319 & .078 & .030 \\ .218 & .167 & .093 \\ .093 & .132 & .108 \end{vmatrix} \qquad C_{\infty}^{12} = \begin{vmatrix} .320 & .079 & .031 \\ .219 & .169 & .096 \\ .095 & .135 & .112 \end{vmatrix}$$

$$C_{63}^{22} = \begin{vmatrix} .323 & .144 & .065 \\ .144 & .156 & .107 \\ .065 & .107 & .101 \end{vmatrix} \qquad C_{\infty}^{22} = \begin{vmatrix} .325 & .146 & .067 \\ .146 & .159 & .111 \\ .067 & .111 & .106 \end{vmatrix}$$

Figure A2.1. Leading principal minors of C_N^{rs} for N = 63 and N = ∞.

In order to obtain upper bounds for the row and column sums of C_∞^{rs} the following Lemma is needed:

Lemma A2.3

Let $$S_i^r = \sum_{j=1}^{\infty} \frac{i_r j_r \sqrt{i_r j_r}}{(i_r^2 + j_r^2)^2} \qquad r=1 \text{ or } 2$$

for some given $i \in (1,2,3\ldots)$.

Then

$$\frac{\pi\sqrt{2}}{16} - \frac{1}{i}\frac{25}{32}\left(\frac{3}{5}\right)^{3/4} \leq S_i^r \leq \frac{\pi\sqrt{2}}{16} + \frac{1}{i}\frac{25}{32}\left(\frac{3}{5}\right)^{3/4}$$

for all i and $r=1$ or 2.

Proof:

$$S_i = \sum_{j=1}^{\infty} \frac{i^{3/2} j^{3/2}}{(i^2+j^2)^2} = \frac{1}{i} \sum_{j=1}^{\infty} \frac{(j/i)^{3/2}}{(1+(j/i)^2)^2} \quad .$$

Let

$$f(x) = \frac{x^{3/2}}{(1+x^2)^2}$$

$$f_{max} = f\left(\sqrt{\frac{3}{5}}\right) = \frac{25}{16}\left(\frac{3}{5}\right)^{3/4} \qquad 0 \leq x \leq \infty \quad .$$

$$\int_0^\infty f(x)dx = \frac{\pi\sqrt{2}}{8}$$

Clearly

$$\lim_{i\to\infty} S_i = \int_0^\infty f(x)dx = \frac{\pi\sqrt{2}}{8} \quad .$$

By considering the discrete sum for finite i it follows that

$$\frac{\pi\sqrt{2}}{8} - \frac{1}{i} f_{max} \leq S_i \leq \frac{\pi\sqrt{2}}{8} + \frac{1}{i} f_{max} \quad .$$

Doing the same analysis for the even sum S_{even}

$$S_{even} = \frac{1}{i} \sum_{j=2,4,..} \frac{(j/i)^{3/2}}{(1+j/i)^2)^2}$$

results in

$$\frac{\pi\sqrt{2}}{16} - \frac{1}{i} g_{max} \leq S_{even} \leq \frac{\pi\sqrt{2}}{16} + \frac{1}{i} g_{max}$$

where the appropriate function is

$$g(x) = \frac{2\sqrt{2}\, x^{3/2}}{(1+4\, x^2)^2} \quad , \int_0^\infty g(x)dx = \frac{\pi\sqrt{2}}{16}$$

and

$$g_{max} = g(\sqrt{\frac{3}{20}}) = f_{max} \quad .$$

Combining these two results proves the Lemma. □

It is now easy to prove Theorem 3.2. As can be easily verified, the row sum

$$\sum_{j=1}^{\infty} C_{1j}^{11} \leq .75853$$

is larger than any other bound that can be obtained for small i (say $i < 20$). Lemma A2.3 shows that this value certainly cannot be exceeded with any larger i. (The factors a_i^{rs} and b_i^{rs} are exponentially small in i and j and present no difficulties. □

Remark:

Computations confirm that the maximum singular value belongs to C^{11}. The upper bound using the matrix C^{21} or C^{12} is

$$\left[(\max_i \sum_{j=1}^{\infty} C_{ij}^{12})(\max_i \sum_{j=1}^{\infty} C_{ij}^{21})\right]^{\frac{1}{2}} \leq .743 \quad .$$

The bound for the matrix C^{22} is given by

$$\max_i \sum_{j=1}^{\infty} C_{ij}^{22} \leq \frac{1}{2}\sqrt{2} \quad \text{(corresponds to } i=\infty \text{ in Lemma A2.3).}$$

Note that the resulting theory also provides lower bounds for the largest singular values. Since the matrices are positive, the smallest row or column sum will be such a bound. (Varga [1962, p. 31]). In particular, computations indicate that $\sigma_{max} > .706$.

The analysis gives an explicit representation for the continuous biharmonic operator in a rectangular region. This representation can be used to study properties of the biharmonic operator in the given geometry.

Finally consider Theorem 3.3. An upper bound for the sum of the singular values of the matrices C^{rs} is needed. Consider the matrix C^{11}. Since C^{11} is symmetric it is sufficient to look at its trace.

$$\sum_{i=1}^{N} C_{ii}^{11} = \sum_{i=1}^{N} \frac{2}{\pi} \frac{1}{i_1} \left(\frac{a_i^{11}}{b_i^{11}}\right)^2 \leq \frac{1}{\pi}(\gamma + \ln N + \delta + O(\tfrac{1}{N}))$$

where γ is Euler's constant, $\gamma = .5772...$ and δ is the contribution from the small term a_i^{11}/b_i^{11}. Letting $N \to \infty$, this shows that the constant in front of the $\ln N$ term in Theorem 3.3 (taken equal to 1 there) tends to $\frac{1}{\pi}$ as N becomes large. A similar argument gives the same result for C^{22}. It is an obvious conjecture that this result is true also for C^{12}, but since it is of little importance in this context the weaker statement in Theorem 3.3 is given instead. This can be proved by considering $\sum_{ij} (C_{ij}^{12})^2$ (the Frobenius norm of C^{12}). □

Figure A2.2 shows the computed sum of the singular values normalized by the factor $\frac{\pi}{\ln N}$ for the three cases of interest, $(C^{21} = (C^{12})^T)$.

N	$\frac{\pi}{\ln N} \sum_{i=1}^{N/2} C_{ii}^{22}$	$\frac{\pi}{\ln N} \sum_{i=1}^{N/2} \sigma_i(C^{12})$	$\frac{\pi}{\ln N} \sum_{i=1}^{N/2} C_{ii}^{11}$
3	.32	.50	1.22
7	.47	.59	1.04
15	.60	.67	.995
31	.67	.73	.985
63	.72	.77	.984
127	.76	.80	.984
255	.79	.83	.986
511	.81	-	.987
1023	.83	-	.988
2047	.85	-	.989
∞	1.0		1.0

Figure A2.2. Normalized sum of singular values.

APPENDIX III

Test problems used in the examples.

The following list defines the test problems refered to by number in the main body of this dissertation. The solution $u(x,y)$ is given. In addition, the subproblems in step 4 of algorithm 4.2, that are nontrivial when solving these problems on the square $0 \leq x,y \leq 1$ are listed, using the notation from section 5.1. This is of importance when considering the results of section 2.3. It also determines the work required to find the solution.

1. $u = xy(1-x)(1-y)$

   Subproblem: (1,1).

   Comments: This problem is frequently used since the truncation error is zero.
   This problem has also been used by Ehrlich and Gupta [1975].

2. $u = x^2 + y^2 - x\, e^x \cos y$

   Subproblems: (1,1), (1,2), (2,1) and (2,2)

   Comments: This problem was considered by Gupta and Manohar [1979], and in a slightly modified form by Ehrlich and Gupta [1975].

3. $u = 2xy + x^3 - 3y^2$

   Subproblems: (1,1), (1,2), (2,1) and (2,2).

   Comments: The problem has zero truncation error if the cubic boundary approximation is used.
   It has been considered by Greenspan and Schultz [1972] and by Gupta and Manohar [1979].

4. $u = x^2y^2(1-x)^2(1-y)^2$

Subproblem: (1,1).

Comments: This solution is simply problem 1 squared. It was used by Bauer and Reiss [1972] and by Gupta and Manohar [1979].

5. $u = (1-\cos 2\pi x)(1-\cos 2\pi y)$

Subproblem: (1,1)

Comments: Another symmetric problem considered by Bauer and Reiss [1972] and by Gupta and Manohar [1979].

6. $u = e^x \sin x + e^y \cos y$

Subproblems: (1,1), (1,2) and (2,1).

Comments: This solution is the sum of two functions each depending on only one variable. Notice that only three subproblems are needed.

7. $u = x^3 \log(1+y) + y/(1+x)$

Subproblems: (1,1), (1,2), (2,1) and (2,2).

Comments: This is a good general problem.

8. $u = e^{\pi(y-1)} \sin \pi x$

Subproblems: (1,1) and (2,1).

Comments: This problem falls in between the highly symmetric and the general problems.

9. $u = \cos \pi x \cos \pi y$

Subproblems: (2,2).

Comments: A highly symmetric problem.

10. $$u = \sum_{p=1}^{8} \sum_{q=1}^{8-p} p y^{p-1} x^{q-1}$$

Subproblems: (1,1), (1,2), (2,1) and (2,2).

Comments: This problem is constructed in order to have a problem where all Taylor coefficients are nonzero up to a total degree of seven.

BIBLIOGRAPHY

J.O. Aasen , On the reduction of a symmetric matrix to tridiagonal form.
BIT Vol. II, 233-242. 1971.

S. Agmon , Lectures on elliptic boundary value problems.
D. Van Nostrand Company, New Jersey. 1965.

N. Aronszajn , R.D. Brown and R.S. Butcher , Construction of the solutions of boundary value problems for the biharmonic operator in a rectangle.
Ann. Inst. Fourier, Grenoble. Vol. 23, 49-89. 1973.

O. Axelson and N. Munksgaard , A class of preconditioned conjugate gradient methods for the solution of a mixed finite element discretization of the biharmonic operator.
Int. J. Num. Meth. Eng. Vol. 14, 1001-1019. 1979.

A. Banerjee , M.B. Banerjee , R. Roy and J.R. Gupta , A generalized biharmonic equation and its applications to hydrodynamic stability.
Jour. Math. Phy. Sci. (India) Vol. 12, 19-33. 1978.

R.E. Bank and D.J. Rose , Marching algorithms for elliptic boundary value problems, I: The constant coefficient case.
SIAM J. Numer. Anal. Vol. 14, 792-829. 1977.

R.E. Bank , A FORTRAN implementation of the generalized marching algorithm.
ACM Trans. Math. Software (TOMS), Vol. 4, 165-176. 1978.

L. Bauer and E.L. Reiss , Block five diagonal matrices and the fast numerical solution of the biharmonic equation.
Math. Comp. Vol. 26, 311-326. 1972.

J.H. Bramble , A second order finite difference analog of the first biharmonic boundary value problem.
Numer. Math. Vol. 9, 236-249. 1966.

C.G. Broyden , A class of methods for solving nonlinear simultaneous equations.
Math. Comp. Vol. 19, 577-593. 1965.

C.G. Broyden , The convergence of a class of double rank minimization algorithms.
J. Inst. Maths. Applics. Vol. 6, 76-90. 1970.

J.R. Bunch and B.N. Parlett , Direct methods for solving symmetric indefinite systems of linear equations.
SIAM J. Numer. Anal. Vol. 8, 639-655. 1971.

O. Buneman , A compact non-iterative Poisson solver.
Report SU-IPR-294. Inst. Plasma Research, Stanford University, 1969.

O. Buneman , Private communication. 1980.

B.L. Buzbee , G.H. Golub and C.W. Nielson , On direct methods for solving Poisson's equation.
SIAM J. Numer. Anal. Vol. 7, 627-656. 1970.

B.L. Buzbee and F.W. Dorr , The discrete solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions.
SIAM J. Numer. Anal. Vol. 11, 753-763. 1974.

B.L. Buzbee , G.H. Golub and J.A. Howell , Vectorization for the Cray-1 of some methods for solving elliptic difference equations.
Proceedings of High speed computer and algorithm organization.
Edited by D.J. Kuck, D.H. Lawrie and A.H. Sameh.
Academic Press New York, 1977.

J.R. Cannon and M.M. Cecchi , The numerical solution of some biharmonic problems by mathematical programming techniques.
SIAM J. Numer. Anal. Vol. 3, 451-466. 1966.

J.R. Cannon and M.M. Cecchi , Numerical experiments on the solution of some biharmonic problems by mathematical programming techniques.
SIAM J. Numer. Anal. Vol. 4, 147-154. 1967.

S. Chakravarthy and D. Anderson , Numerical conformal mapping.
Math. Comp. Vol. 33, 953-969. 1979.

S. Christiansen and P. Hougaard , An investigation of a pair of integral equations for the biharmonic problem.
J. Inst. Maths. Applics. Vol. 22, 15-27. 1978.

P.G. Ciarlet , The finite element method for elliptic problems.
North-Holland, Amsterdam 1978.

A.K. Cline , Several observations on the use of conjugate gradient methods.
ICASE report no. 76-22. 1976.
NASA Langley research center, Hampton, Virginia.

C.V. Coffman , On the structure of solutions to $\Delta^2 u = \lambda u$ which satisfy the clamped plate conditions on a right angle. (Manuscript)
Department of Mathematics Carnegie-Mellon University 1980.

L. Collatz , Numerische Behandlung von Differentialgleichungen.
Springer Verlag. Berlin 1955.

P. Concus , G.H. Golub and D.P. O'Leary , A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations.
Proc. Symposium on sparse matrix computations
Edited by J.R. Bunch and D.J. Rose
Academic Press, New York 1976.

S.D. Conte and R.T. Dames , On an alternating direction method for solving the plate problem with mixed boundary conditions.
J. Assoc. Comput. Mach. Vol. 7, 264-273. 1960.

R. Courant , K.O. Friedrichs and H. Lewy , Über die partiellen Differenzengleichungen der matematischen Physik.
Math. Ann. Vol. 100, 32-74. 1928.

G. Dahlquist and Å. Björck, Numerical methods.
Prentice Hall, New Jersey 1974.

N. Dessi and M.G. Manca , Solution of the biharmonic equation by linear programming methods.
CALCOLO (Italy) Vol. 13, 109-121. 1976.

E. Detyna , Point cyclic reductions for elliptic boundary-value problems. I. The constant coefficient case.
J. Comp. Phys. Vol. 33, 204-216. 1979.

N. Distéfano , Dynamic programming and the solution of the biharmonic equation.
Int. J. Num. Meth. Eng. Vol. 3, 199-213. 1971.

J.J. Dongarra , J.R. Bunch , C.B. Moler and G.W. Stewart , LINPACK users' guide.
Argonne National Laboratory. 1979.

L.W. Ehrlich , Solving the biharmonic equation as coupled finite difference equations.
SIAM J. Numer. Anal. Vol. 8, 278-287. 1971.

L.W. Ehrlich , Coupled harmonic equations, SOR and Chebyshev accelleration.
Math. Comp. Vol. 26, 335-343. 1972.

L.W. Ehrlich , Solving the biharmonic equation in a square: A direct versus a semidirect method.
Comm. ACM. Vol. 16, 711-714. 1973.

L.W. Ehrlich and M.M. Gupta , Some difference schemes for the biharmonic equation.
SIAM J. Numer. Anal. Vol. 12, 773-790. 1975.

H.E. Fettis , Private communication. 1979.

G. Fichera , Sul calcolo degli autovalori della piastra quadrata incastrata lungo il bordo.
Lincei-Rendiconti Science fisiche, matematiche e naturali. Vol. 40, 725-733. 1966.

B. Fornberg , A numerical method for conformal mapping.
(Manuscript) 1980.

D. Gaier , Knostruktive Methoden der Konformen Abbildung.
Springer tracts in natural philosophy. Vol. 3 Springer, Berlin. 1964.

D.M. Gay , Some convergence properties of Broydens method.
SIAM J. Numer. Anal. Vol. 16, 623-630. 1979.

R. Glowinski , Approximations externes, par elements finis de Lagrange d'ordre un et deux, du problème de Dirichlet pour l'operateur biharmonique. Methode iterative de resolution des problemes approches.
Topics in numerical analysis, proceedings of the royal Irish academy conference on numerical analysis, 1972. Edited by John J.H. Miller. Academic Press, London 1973.

R. Glowinski , J.L. Lions and R. Tremolieres , Analyse Numérique des Inéquations Variationelles, Vol. 1. Dunod-Bordas, Paris 1976.

R. Glowinski and O. Pironneau , Numerical methods for the first biharmonic equation and for the two-dimensional Stokes problem.
SIAM Review. Vol. 21, 167-212. 1979.

G.H. Golub , An algorithm for the discrete biharmonic equation.
Unpublished, see the appendix of L.W. Ehrlich 1973 .

E. Goursat , Sur l'équation $\Delta\Delta u = 0$.
Bull de la Soc. Math. de France. Vol. 26, 236-237. 1898.

I.S. Gradshteyn and I.M. Ryzhik, Table of Integrals, Series and Products.
Academic Press, New York 1965.

A. Greenbaum , Comparison of splittings used with the conjugate gradient algorithm.
Numer. Math. Vol. 33, 181-194. 1979.

D. Greenspan and D. Schultz , Fast finite difference solution of biharmonic problems.
Comm. ACM. Vol. 15, 347-350. 1972.

M.M. Gupta , Discretization error estimates for certain splitting procedures for solving first biharmonic boundary value problems.
SIAM J. Numer. Anal. Vol. 12, 364-377. 1975.

M.M. Gupta and R.P. Manohar , Direct solution of the biharmonic equation using noncoupled approach.
J. Comp. Phys. Vol. 33, 236-248. 1979.

M.H. Gutknecht , Solving Theodorsen's integral equation for conformal maps with the fast Fourier transform I.
(Manuscript) 1979.

P. Henrici , Fast Fourier methods in computational complex analysis.
SIAM Review. Vol. 21, 481-527. 1979.

M.R. Hestenes and E. Stiefel , Methods of conjugate gradients for solving linear systems.
Nat. Bur. Standards, J. of Research. Vol. 49, 409-436. 1952.

R.W. Hockney , A fast direct solution of Poisson's equation using Fourier analysis.
J. Assoc. Comput. Mach. Vol. 12, 95-113. 1965.

D.A. Jacobs , The strongly implicit procedure for biharmonic problems.
J. Comp. Phys. Vol. 13, 303-315. 1973.

A. Jennings , Influence of the eigenvalue spectrum on the convergence rate of the conjugate gradient method.
J. Inst. Maths. Applics. Vol. 20, 61-72. 1977.

A.I. Kalandiya , Mathematical methods of two-dimensional elasticity.
Mir publishers, Moscow 1975.

S. Kaniel , Estimates for some computational techniques in linear algebra.
Math. Comp. Vol. 20, 369-378. 1966.

L.V. Kantorovich and V.I. Krylov , Approximate methods of higher analysis.
P. Noordhoff, Groningen 1958.

J.T. Katsikadelis , An integral equation solution of the plane problem of the theory of elasticity.
Mech. Res. Comm. Vol. 4, 199-208. 1977.

D.R. Kincaid and D.M. Young , Adapting iterative algorithms developed for symmetric systems to nonsymmetric systems.
To appear in proceedings from Elliptic problem solvers conference. Editor M. Schultz. Academic Press. 1980.

V.D. Kupradze , Potential methods in the theory of elasticity.
Israel program for scientific translations. Jerusalem 1965.
Translated by H. Gutfreund and published in the USA by Daniel Davey, New York.

J.R. Kuttler , A finite difference approximation for the eigenvalues of the clamped plate.
Numer. Math. Vol. 17, 230-238. 1971.

L.D. Landau and E.M. Lifchitz , Fluid mechanics.
Pergamon Press, London 1959.

L.D. Landau and E.M. Lifchitz , Theory of elasticity.
Pergamon Press, London 1970.

J.L. Lions and E. Magenes , Non-homogeneous boundary value problems and applications I.
Springer Verlag, New York 1972.

D.G. Luenberger , Introduction to linear and nonlinear programming.
Addison-Wesley, Massachusetts 1973.

MACSYMA , Reference manual. Version 9, July 1977.
The Mathlab group, Laboratory for computer science, MIT.

W. Magnus and F. Oberhettinger , Formeln und Sätze für die speziellen Funktionen der mathematischen Physik.
Springer Verlag, Berlin 1948.

J.W. McLaurin , Boundary eigenvalues of clamped plates.
Journal of applied mathematics and physics (ZAMP). Vol. 19, 676-681. 1968.

J.W. McLaurin , A general coupled equation approach for solving the biharmonic boundary value problem.
SIAM J. Numer. Anal. Vol. 11, 14-33. 1974.

C. Miranda , Formule di maggiorazione e teorema di esistenza per le funzioni biarmoniche di due variabili.
Giorn. Mat. Battaglini. Vol. 78, 97-118. 1948.

N. Munksgaard , Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients.
ACM Trans. Math. Soft. Vol. 6, 206-219. 1980.

N.I. Muskhelishvili , Some basic problems of the mathematical theory of elasticity.
P. Noordhoff, Groningen 1963.

L. Nazareth , A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms.
SIAM J. Numer. Anal. Vol. 16, 794-800. 1979.

J. Nocedal , Updating quasi-Newton matrices with limited storage.
IIMAS, Universidad Nacional Autónoma de Mexico, 1979.

J.M. Ortega and W.C. Rheinboldt , Iterative solution of nonlinear equations in several variables.
Academic Press, New York 1970.

C.C. Paige and M.A. Saunders , Solution of sparse indefinite systems of linear equations.
SIAM J. Numer. Anal. Vol. 12, 617-629. 1975.

B.N. Parlett , A new look at the Lanczos algorithm for solving symmetric systems of linear equations.
Lin. Alg. App. Vol. 29, 323-346. 1980.

S.V. Parter , On two line iterative methods for the Laplace and biharmonic difference equations.
Numer. Math. Vol. 1, 240-252. 1959.

W. Proskurowski and O. Widlund , On the numerical solution of Helmholtz's equation by the capacitance matrix method.
Math. Comp. Vol. 30, 433-468. 1976.

W. Proskurowski , Four FORTRAN programs for numerically solving Helmholtz's equation in an arbitrary bounded planar region.
Lawrence Berkeley Laboratory, Report LBL-7516. 1978.

W. Proskurowski and O. Widlund , A finite element-capacitance matrix method for the Neuman problem for Laplace's equation.
To appear. 1980.

M. Rahman and R.A. Usmani , A note on the solution of the biharmonic equation arising in plate deflection theory.
J. Phys. Soc. Japan. Vol. 43, 698-700. 1977.

K. Rektorys , The method of least squares on the boundary and very weak solutions of the first biharmonic problem.
EQUADIFF IV 1979. Lecture notes in mathematics Vol. 703, 348-355.
Springer Verlag, Berlin 1979.

G.R. Richter , An integral equation method for the biharmonic equation.
Proceedings from Advances in computer methods for partial differential equations II. R. Vichnevetsky (editor), Publ. IMACS (AICA), 41-45. 1977.

J.B. Rosser , Majorization formulas for a biharmonic function of two variables.
SIAM J. Numer. Anal. Vol. 17, 207-220. 1980.

A.H. Sameh , S.C. Chen and O.J. Kuck , Parallel Poisson and biharmonic solvers.
Computing. Vol. 17, 219-230. 1976.

J. Schröder , U. Trottenberg and K. Witsch , On fast Poisson solvers and applications.
Proceedings of a conference on numerical treatment of differential equations. Lecture notes in mathematics Vol. 631, 153-187.
Springer Verlag, Berlin 1978.

A.H. Sherman , On the efficient solution of sparse systems of linear and nonlinear equations.
Ph.D. Thesis, Department of Computer Science, Yale University. 1975.

V.G. Sigillito , A priori inequalities and approximate solutions of the first boundary value problem for $\Delta^2 u = f$.
SIAM J. Numer. Anal. Vol. 13, 251-260. 1976.

J. Smith , The coupled equation approach to the numerical solution of the biharmonic equation by finite differences, I.
SIAM J. Numer. Anal. Vol. 5, 323-339. 1968.

J. Smith , The coupled equation approach to the numerical solution of the biharmonic equation by finite differences, II.
SIAM J. Numer. Anal. Vol. 7, 104-111. 1970.

J. Smith , On the approximate solution of the first boundary value problem for $\nabla^4 u = f$.
SIAM J. Numer. Anal. Vol. 10, 967-982. 1973.

S.L. Sobolev , On estimates for certain sums for functions defined on a grid.
Izv. Akad. Nauk SSSR, Ser. Mat. Vol. 4, 5-16. 1940.

I.S. Sokolnikoff , Mathematical theory of elasticity.
McGraw Hill, New York 1946.

G.W. Stewart , The convergence of the method of conjugate gradients at isolated extreme points of the spectrum.
Numer. Math. Vol. 24, 85-93. 1975.

G. Strang and G.J. Fix , An analysis of the finite element method.
Prentice Hall, New Jersey 1973.

P.N. Swarztrauber and R. Sweet , Efficient FORTRAN subprograms for the solution of elliptic partial differential equations.
NCAR-TN/IA-109. National Center for Atmospheric Research, Boulder, Colorado. 1975.

P.N. Swarztrauber , The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle.
SIAM Review Vol. 19, 490-501. 1977.

P.N. Swarztrauber , A package of FORTRAN subprograms for the fast Fourier transform of periodic and other symmetric sequences.
Version 2 February 1978. National Center for Atmospheric Research Boulder, Colorado.

G.T. Symm , An integral equation method in conformal mapping.
Numer. Math. Vol. 9, 250-258. 1966.

W.P. Tang , On some new difference schemes for elliptic differential equations.
Masters thesis, Department of mathematics, Futan University Shanghai. 1964.

G.J. Tee , A novel finite difference approximation to the biharmonic operator.
The computer journal. Vol. 6, 177-192. 1963.

R. Temam , Navier Stokes equations.
North-Holland, Amsterdam 1977.

C. Temperton , Direct methods for the solution of the discrete Poisson equation: Some comparisons.
J. Comp. Phys. Vol. 31, 1-20. 1979.

C. Temperton , On the FACR(l) algorithm for the discrete Poisson equation.
J. Comp. Phys. Vol. 34, 314-329. 1980.

L.N. Trefethen , Numerical computation of the Schwarz-Christoffel transformation.
SIAM J. Sci. Stat. Comput. Vol. 1, 82-102. 1980.

A.N. Tychonoff and A.A. Samarski , Differentialgleichungen der Matematischen Physik.
VEB Deutscher Verlag der Wissenschaften, Berlin 1959.

R.R. Underwood , An iterative block Lanczos method for the solution of large sparse symmetric eigenproblems.
Ph.D. Thesis. Stanford University. 1975.

M. Vajteršic , A fast algorithm for solving the first biharmonic boundary value problem.
Computing Vol. 23, 171-178. 1979.

R.S. Varga , Matrix iterative analysis.
Prentice-Hall, New Jersey 1962.

H. Vaughan , Series solution of the biharmonic equation in the rectangular domain with some applications to mechanics.
Proc. Camb. Phil. Soc. Vol. 76, 563-585. 1974.

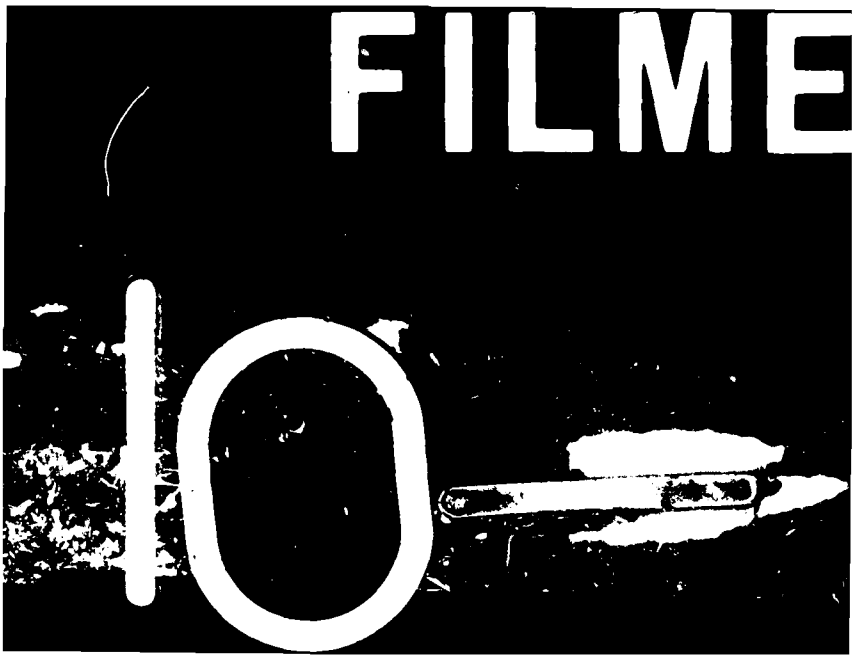
J.H. Wilkinson , The algebraic eigenvalue problem.
Oxford University Press, London 1965.

D.M. Young , Iterative solution of large linear systems.
Academic Press, New York 1972.

O.C. Zienkiewicz , The finite element method.
Third edition, McGraw Hill, London 1977.

M. Zlámal , Discretization and error estimates for elliptic boundary value problems of the fourth order.
SIAM J. Numer. Anal. Vol. 4, 626-639. 1967.

R. Zurmühl , Behandlung der Plattenaufgabe nach dem verbesserten Differenzenverfahren
Z. angew. Math. Mech. Vol. 37, 1-16. 1957.


FILME
10