QUADRATIC SPLINE COLLOCATION FOR POISSON'S AND BIHARMONIC EQUATIONS IN THE UNIT SQUARE

by
Jonathan Maack

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematical and Computer Sciences).

Golden, Colorado

Date 1/12/09

Signed: *Jonathan Maack*

Jonathan A. Maack

Signed: *Graeme Fairweather*

Dr. Graeme Fairweather

Thesis Advisor

Golden, Colorado

Date 1/12/09

Signed: *D P Mehta*

Dr. Dinesh Mehta

Professor and Head

Department of Mathematical

and Computer Sciences

ABSTRACT

We develop new quadratic spline collocation (QSC) methods for the solution of Poisson's and biharmonic equations in the unit square, $\Omega = (0,1) \times (0,1)$. In these methods, the differential operator and the right hand side of the partial differential equation are perturbed to obtain the collocation equations. The collocation points are taken to be the midpoints of the cells of a uniform partition of Ω on which the spline space is defined.

We derive a new QSC method for solving Poisson's equation with Dirichlet boundary conditions: $\Delta u = f$ in Ω and $u = 0$ on $\partial\Omega$, where Δ denotes the standard Laplacian operator and $\partial\Omega$ is the boundary of Ω. We derive a matrix decomposition algorithm (MDA) which employs fast Fourier transforms to solve the resulting linear system at a cost of $O(N^2 \log N)$ operations on an $N \times N$ partition of Ω. We also prove the existence and uniqueness of the solution. Finally, we present numerical results that indicate optimal global accuracy and superconvergence phenomena.

We extend the method derived for Poisson's equation to the solution of the biharmonic Dirichlet problem: $\Delta^2 u = f$ in Ω and $u = \partial u/\partial n = 0$ on $\partial\Omega$, where $\partial/\partial n$ is the outward normal derivative on $\partial\Omega$. We solve the biharmonic problem by rewriting it as a coupled system of two second-order partial differential equations in u and $v \equiv \Delta u$. Using the Schur complement approach, we show that the resulting linear system can be solved at a cost of $O(N^3 \log N)$ operations. Finally, we present numerical results that indicate global optimal accuracy and superconvergence phenomena.

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1

INTRODUCTION

The main goal of this thesis is to develop new optimal quadratic spline collocation (QSC) methods for the solution of two problems in the unit square. The first problem is Poisson's equation subject to homogeneous Dirichlet boundary conditions:

$$\Delta u = f \ \text{ in } \ \Omega, \qquad u = 0 \ \text{ on } \ \partial\Omega, \tag{1.1}$$

where Δ denotes the standard Laplacian operator, $\Omega = (0,1) \times (0,1)$ is the unit square, and $\partial\Omega$ denotes its boundary. Poisson's equation is most frequently encountered in electrostatics, mechanics, physics of conducting media, and the description of a gravitational potential. In these problems, the variation of the field characterized by either an electric or gravitational potential associated with a spatial distribution leads to Poisson's equation.

The second problem is the biharmonic Dirichlet problem,

$$-\Delta^2 u = -f \ \text{ in } \ \Omega, \qquad u = 0 \ \text{ on } \ \partial\Omega, \ \ \frac{\partial u}{\partial n} = 0 \ \text{ on } \ \partial\Omega, \tag{1.2}$$

where $\partial u/\partial n$ denotes the outward normal derivative on $\partial\Omega$. This problem models bending of a thin elastic clamped rectangular plate, equilibrium of an elastic rectangle and flow of very viscous fluid in a square cavity [23]. A discussion of the history of problem (1.2) can also be found in [23].

Spline collocation methods for the solution of boundary value problems for ordinary and partial differential equations have been frequently used over the past several decades. In these methods, the boundary conditions and differential equation are satisfied at specific points of a partition of the problem domain. However, as demonstrated by de Boor [14], the basic spline collocation methods are suboptimal. In fact, the basic nodal cubic spline collocation (NCSC) method is only second-order accurate. For a spline space of degree k, we expect an optimal method to produce accuracy of order $k+1$. Hence, we would expect fourth-order global accuracy from an NCSC method and third-order global accuracy from a QSC method.

The suboptimal performance led to research into optimal cubic spline collocation methods. De Boor and Swartz [15] developed orthogonal spline collocation (OSC) for two-point boundary-value problems (TPBVPs), in which optimal convergence rates were obtained by collocating at Gauss points. For TPBVPs, researchers succeeded in modifying basic NCSC to obtain optimal accuracy with a uniform partition. Fyfe [18] developed a two step NCSC method (TSM) in which one first determines the cubic spline solution by collocating at the nodal points of the partition then improves it by using a deferred correction like approach. Archer [5, 6] and Daniel and Swartz [16] independently developed a one step NCSC method (OSM) for TPBVPs where the solution is directly determined by using a high order perturbation of the differential equation. Taking a comparable approach, Houstis et al., [20] developed an optimal method using QSC instead of cubics for second-order TPBVPs. In this method, the differential equation is collocated at the midpoints of the subintervals of the partition on which the spline space is defined.

Houstis et al., [21] generalized Fyfe's TSM and Archer's OSM to elliptic partial differential equations in the unit square with a focus on Dirichlet and Neumann Helmholtz problems. However, it has been shown recently that the analysis of the OSM for the

Dirichlet problem in [21] is incorrect [1, 3]. Furthermore, the OSM is not optimally fourth-order accurate for Neumann boundary conditions as claimed in [21]. Since Archer's OSM when applied to TPBVPs with Neumann, mixed or periodic boundary conditions is only third-order accurate, which is suboptimal, optimal accuracy is not expected when extended to elliptic problems; see, for example, [24].

Christara [13] extended the work of Houstis et al., [20] to second order linear elliptic problems with Dirichlet and Neumann boundary conditions using biquadratic collocation methods. These problems are of the form

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu = g, \quad \text{in} \quad \Omega, \tag{1.3}$$

with the mixed boundary conditions

$$\alpha u + \beta \frac{\partial u}{\partial n} = \gamma, \quad \text{on} \quad \partial\Omega, \tag{1.4}$$

where $u, a, b, c, d, e, f, g, \alpha, \beta, \gamma$ are functions of x and y. Note that (1.1) is of this form with $a = c = \alpha = 1$ and $b = d = e = f = \beta = \gamma = 0$. The method developed in [13] is globally optimal and superconvergent, that is, the method exhibits $O(h^{3-k})$ accuracy globally for the kth partial derivative, and the method exhibits fourth-order accuracy at the nodal and collocation points of the u approximation, third-order accuracy at the Gauss points of the u_x and u_y approximations, and second-order accuracy at the collocation points of the u_{xx} and u_{yy} approximations [13].

The QSC, OSC, and NCSC methods produce large systems of equations and require efficient schemes for their solution. Frequently, these methods involve matrix decomposition algorithms (MDAs). In these algorithms, fast Fourier transforms (FFTs) are used to directly solve the resulting linear systems at a cost of $O(N^2 \log N)$ operations on an $N \times N$ uniform partition. In [13], Christara does not formulate an MDA for the

OSM. For the new QSC methods presented here, MDAs are formulated, resulting in tridiagonal or, at worst, banded systems of equations.

Bialecki et al., [8] developed and implemented MDAs for the collocation equations of [21]. Specifically, MDAs are formulated for the optimal OSM with Dirichlet boundary conditions and the optimal and superconvergent TSM with Dirichlet, Neumann, mixed, and periodic boundary conditions. In [9], Bialecki et al., developed new optimal OSMs for the four types of boundary conditions, and for each of the resulting linear systems, an MDA is formulated. These new OSMs are constructed by judiciously perturbing the differential operator and the right hand side of the differential equation.

In [24], Nguyen, developed a new one-step modified QSC method for the Helmholtz equation with Dirichlet, Neumann, mixed and periodic boundary conditions; see also [11]. In each method, Nguyen perturbs the differential operator and the right hand side of the differential equation to obtain an optimal superconvergent method, and then uses an MDA to efficiently solve the resulting linear system. In [10], the new QSC method of [11, 24] is extended to the solution of elliptic partial differential equations of the form (1.3) with inhomogeneous boundary conditions, where $a = 1$, $b = d = 0$ and c, e, f are functions of y.

The most frequently used methods for solving (1.2) are finite difference and finite element methods. These methods use two different approaches to solve (1.2). In the first approach, the direct approach, (1.2) is discretized immediately using, for example, a thirteen-point finite difference formula [12, 19] or C^1 Hermite bicubic finite elements [26]. The second approach is the mixed approach, in which (1.2) is first reduced to a coupled system of two second-order partial differential equations in u and $v \equiv \Delta u$. This approach has the advantage of producing an approximation to u as well as Δu, which often has physical meaning such as vorticity in fluid dynamics. The coupled system is then discretized using, for example, finite difference [4, 17] or finite element [22, 25] methods. The algorithms in [12, 22] use MDAs to implement their methods at a cost of $O(N^2 \log N)$.

Recently, Bialecki [7] used the mixed approach and OSC to solve (1.2). In this work, Bialecki used the Schur complement method, preconditioned conjugate gradient method and FFTs to efficiently solve the resulting linear system of equations. The resulting algorithm can be performed at a cost of $O(N^2 \log N)$ arithmetic operations.

Furthermore, problems (1.1) and (1.2) were considered by Abushama in [1, 2, 3]. In these works, optimal modified NCSC methods are formulated by, again, perturbing the differential operator and the right hand side of the differential equations. This method also exhibited fourth-order superconvergent results at the nodal points for the approximation to the first partial derivative and the cross derivative. In the case of problem (1.1), they use an MDA to solve the resulting linear system [1, 3]. For problem (1.2), Abushama and Bialecki [1, 2] use the Schur complement approach, the pre-conditioned conjugate gradient method, and an MDA to efficiently solve the linear system.

This thesis closely parallels Abushama's work [1], with quadratic splines replacing cubic splines. Quadratic spline collocation methods are advantageous because no corner or special boundary equations are required. In particular, when solving the resulting systems, all coefficients are determined simultaneously, unlike the cubic case where it is necessary to solve for the boundary coefficients first. We also see a difference from the cubic case as Neumann boundary conditions require a perturbation in the quadratic case. Globally, the cubic method is more accurate; however, the QSC methods presented here produce accuracy comparable to the cubic case at the nodal points.

A brief outline of this thesis is as follows. In Chapter 2, we present definitions and properties used throughout this work.

In Chapter 3, we derive the new QSC method for solving a second-order TPBVP based on the approach of Houstis et al., [20] and Christara [13]. In this method, the right hand side of the collocation equations are perturbed producing a third-order accurate method but preserving the basic method's tridiagonal system.

In Chapter 4, we extend this method to solve (1.1). Numerical results are presented to show that the method exhibits optimal convergence globally and is superconvergent at specific points. Furthermore, we describe an MDA that produces a solution at a cost of $O(N^2 \log N)$ operations.

In Chapter 5, we further develop the new QSC method to the solution of a fourth-order TPBVP. Here, the problem first is rewritten as a coupled system of two second-order TPBVPs and then discretized. In this case, we must derive a perturbation of the Neumann boundary conditions to preserve optimal accuracy. We prove the existence and uniqueness of the solution. Finally, numerical results are presented to demonstrate the method's accuracy. For comparison purposes, a single test problem of the method without perturbed boundary conditions is presented.

In Chapter 6, we solve a problem closely related to (1.2) by combining the methods of Chapters 4 and 5. In an approach similar to that in Chapter 5, we rewrite the fourth-order partial differential equation as a coupled system of two second-order partial differential equations in u and Δu. An MDA is formulated to solve the resulting linear system. Again, we show the solution of the method exists and is unique. Finally, numerical results are given to support the accuracy of the method.

In chapter 7, we solve (1.2). We again use the mixed approach and rewrite the problem as a system in u and Δu and then combine the methods of Chapters 4 and 5. We show how to solve the resulting linear system efficiently using the Schur complement approach and the MDA developed in Chapter 6. Then we present numerical results to demonstrate the accuracy of the method.

In Chapter 8, we summarize this work and highlight a few areas of potential future research.

Chapter 2

PRELIMINARIES

There are several commonalities shared by all of the problems discussed. First, we define $\rho_x = \{x_i\}_{i=0}^N$ to be a uniform partition of $I = [0, 1]$ such that $x_i = ih, \quad i = 0, 1, \ldots, N$, where $h = 1/N$. We define the collocation points $\{\tau_i\}_{i=1}^N$ as

$$\tau_i = (x_i + x_{i-1})/2, \tag{2.1}$$

which are the midpoints of the subintervals $I_i = [x_{i-1}, x_i]$, $i = 1, \ldots, N$.

Let S_2 be the space of quadratic splines defined by

$$S_2 = \{v : v \in C^1[0, 1], v|_{I_i} \in P_2, 1 \leq i \leq N\}, \tag{2.2}$$

where P_2 is the set of polynomials of degree ≤ 2. In order to define a basis for S_2, we extend the partition ρ_x by defining $x_i = ih, \quad i = -2, -1, N + 1, N + 2$. We take a basis of S_2 to be $\{\mathcal{B}_m\}_{m=0}^{N+1}$, where

$$\mathcal{B}_m(x) = \frac{1}{2}\xi\left(\frac{x}{h} - m + 2\right), \tag{2.3}$$

and ξ is the quadratic spline function defined by

$$\xi(x) = \begin{cases} x^2, & x \in [0, 1], \\ -3 + 6x - 2x^2, & x \in [1, 2], \\ 9 - 6x + x^2, & x \in [2, 3], \\ 0, & \text{otherwise}. \end{cases} \tag{2.4}$$

Using (2.3) and (2.4), we get

$$\mathcal{B}_m(\tau_i) = \begin{cases} 1/8, & i = j \pm 1, \\ 3/4, & i = j, \\ 0, & \text{otherwise}, \end{cases} \qquad \mathcal{B}'_m(\tau_i) = \begin{cases} \mp 1/(2h), & i = j \pm 1, \\ 0, & \text{otherwise}, \end{cases}$$

$$\mathcal{B}''_m(\tau_i) = \begin{cases} 1/h^2, & i = j \pm 1, \\ -2/h^2, & i = j, \\ 0, & \text{otherwise}. \end{cases} \tag{2.5}$$

We define the space S_2^D by

$$S_2^D = \{v \in S_2 : v(0) = v(1) = 0\},$$

that is, the elements of S_2^D satisfy homogeneous Dirichlet boundary conditions. As a basis for S_2^D, we chose $\{\mathcal{B}_m^D\}_{m=1}^N$, where

$$\mathcal{B}_m^D = \begin{cases} \mathcal{B}_m - \mathcal{B}_{m-1}, & m = 1, \\ \mathcal{B}_m, & m = 2, \ldots, N-1, \\ \mathcal{B}_m - \mathcal{B}_{m+1}, & m = N. \end{cases} \tag{2.6}$$

Thus, from (2.5) and (2.6), we have

$$\mathcal{B}_m^D(\tau_i) = \begin{cases} 5/8, & i = j = 1, \text{ or } i, j = N, \\ 1/8, & i = 1, 2, \ldots, N; \;\; j = i \pm 1, \\ 3/4, & i = j = 2, 3, \ldots, N-1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

$$[\mathcal{B}_m^D]''(\tau_i) = \begin{cases} -3/h^2, & i = j = 1, \text{ or } i = j = N, \\ 1/h^2, & i = 1, 2, \ldots, N; \;\; j = i \pm 1, \\ -2/h^2, & i = j = 2, 3, \ldots, N, \\ 0, & \text{otherwise.} \end{cases}$$

Next, let $w_h \in S_2$ denote the interpolant of Houstis et al., [20], defined by

$$\begin{cases} w_h(\tau_i) = u(\tau_i), \quad i = 1, \ldots, N, \\ w_h(0) = u(0) - \dfrac{h^4}{128} u^{(4)}(0), \\ w_h(1) = u(1) - \dfrac{h^4}{128} u^{(4)}(1). \end{cases} \tag{2.8}$$

Then [20, (6b)], we have

$$w_h''(\tau_i) = u''(\tau_i) - \frac{h^2}{24} u^{(4)}(\tau_i) + O(h^4), \quad i = 1, \ldots, N, \tag{2.9}$$

and from [20, (20)], we have

$$\begin{cases} w_h'(0) = u'(0) - \dfrac{h^2}{12}u^{(3)}(0) + O(h^4), \\ \\ w_h'(1) = u'(1) - \dfrac{h^2}{12}u^{(3)}(1) + O(h^4). \end{cases} \tag{2.10}$$

Next, we introduce the matrices T, Λ_T, and Z, which are defined by

$$T = \begin{bmatrix} -3 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -3 \end{bmatrix}_{N\times N} \tag{2.11}$$

$$\Lambda_T = \mathrm{diag}(\lambda_i^T)_{i=1}^N, \quad \text{where } \lambda_i^T = -4\sin^2\left(\frac{i\pi}{2N}\right), \tag{2.12}$$

and

$$Z = [z_{i,j}]_{i,j=1}^N, \tag{2.13}$$

where

$$z_{i,j} = \begin{cases} \sqrt{\dfrac{2}{N}}\sin\dfrac{(2i-1)j\pi}{2N}, & i = 1,\ldots,N; \quad j = 1,\ldots,N-1, \\ \\ \sqrt{\dfrac{1}{N}}(-1)^{i-1}, & i = 1,\ldots,N; \quad j = N. \end{cases} \tag{2.14}$$

Now, the diagonal elements of the matrix Λ_T are the eigenvalues of T and the columns of the matrix Z are the corresponding eigenvectors of T. It follows that

$$Z^T T Z = \Lambda_T, \tag{2.15}$$

and

$$Z^T Z = Z Z^T = I, \tag{2.16}$$

where I is the $N \times N$ identity matrix.

We define the tensor product, denoted by $\otimes$, of two matrices as follows: for the $M \times N$ matrix $A = (a_{i,j})$ and $P \times Q$ matrix B, then the matrix $A \otimes B$ is the $MP \times NQ$ block matrix whose (i,j) block is $(a_{i,j})B$.

The last property presented in this chapter is used throughout this thesis. Let $\mathcal{I}, \mathcal{J}, \mathcal{M}, \mathcal{N}$ be finite sets of increasing indices. We assume without loss of generality that

$$\mathcal{I} = \{1, \ldots, I'\}, \quad \mathcal{J} = \{1, \ldots, J'\}, \quad \mathcal{M} = \{1, \ldots, M'\}, \quad \mathcal{N} = \{1, \ldots, N'\}.$$

Then the matrix-vector form of

$$\phi_{i,j} = \sum_{m \in \mathcal{M}} c^{(1)}_{i,m} \sum_{n \in \mathcal{N}} c^{(2)}_{j,n} \psi_{m,n}, \quad i \in \mathcal{I}, \quad j \in \mathcal{J}, \tag{2.17}$$

is

$$\boldsymbol{\phi} = (C_1 \otimes C_2) \boldsymbol{\psi}, \tag{2.18}$$

where

$$C_1 = \left(c^{(1)}_{i,m}\right)_{i \in \mathcal{I}, m \in \mathcal{M}}, \quad C_2 = \left(c^{(2)}_{j,n}\right)_{j \in \mathcal{J}, n \in \mathcal{N}},$$

and

$$\boldsymbol{\phi} = [\phi_{1,1}, \ldots, \phi_{1,J'}, \ldots, \phi_{I',1}, \ldots, \phi_{I',J'}]^T,$$

$$\boldsymbol{\psi} = [\psi_{1,1}, \ldots, \psi_{1,N'}, \ldots, \psi_{M',1}, \ldots, \psi_{M',N'}]^T.$$

Chapter 3

SECOND-ORDER TWO-POINT BOUNDARY VALUE PROBLEM

In this chapter, we consider the two-point boundary-value problem (TPBVP)

$$u''(x) + cu(x) = f(x), \qquad x \in [0, 1], \tag{3.1}$$

where $c < 0$ and

$$u(0) = u(1) = 0. \tag{3.2}$$

The method developed in this chapter is generalized in Chapter 4 to solve (1.1) and used in Chapter 5 in the solution of a fourth-order TPBVP.

3.1 Derivation

To derive our method, we use the results of Houstis et al., [20] described in equations (2.8)–(2.10). Combining (2.8), (2.9) and (3.1), we have, for $1 \le i \le N$,

$$w_h''(\tau_i) + cw_h(\tau_i) = u''(\tau_i) - \frac{h^2}{24}u^{(4)}(\tau_i) + O(h^4) + cu(\tau_i). \tag{3.3}$$

Upon using (3.1), equation (3.3) becomes

$$w_h''(\tau_i) + cw_h(\tau_i) = f(\tau_i) - \frac{h^2}{24}u^{(4)}(\tau_i) + O(h^4). \tag{3.4}$$

Now, rearranging (3.1) and differentiating the equation twice and using (3.1), we have

$$u^{(4)}(\tau_i) = f''(\tau_i) - cu''(\tau_i) = f''(\tau_i) - c[f(\tau_i) - cu(\tau_i)],$$

and substituting into (3.4) we have

$$w_h''(\tau_i) + cw_h(\tau_i) = f(\tau_i) - \frac{h^2}{24}[f''(\tau_i) - cf(\tau_i) + c^2 w_h(\tau_i)] + O(h^4). \tag{3.5}$$

Setting

$$\kappa = \left(1 + \frac{ch^2}{24}\right), \tag{3.6}$$

equation (3.5) becomes

$$w_h''(\tau_i) + c\kappa w_h(\tau_i) = \kappa f(\tau_i) - \frac{h^2}{24} f''(\tau_i) + O(h^4). \tag{3.7}$$

We define our approximation $u_h \in S_2^D$ by

$$u_h''(\tau_i) + c\kappa u_h(\tau_i) = \kappa f(\tau_i) - \frac{h^2}{24} f''(\tau_i), \quad i = 1, \ldots, N, \tag{3.8}$$

which is obtained by replacing w_h with u_h in (3.7) and dropping the $O(h^4)$ term. Now let

$$u_h(x) = \sum_{m=1}^{N} u_m \mathcal{B}_m^D(x). \tag{3.9}$$

Then substituting (3.9) into (3.8), we have

$$\sum_{m=1}^{N} u_m \mathcal{B}_m^{D\prime\prime}(\tau_i) + c\kappa \sum_{m=1}^{N} u_m \mathcal{B}_m^D(\tau_i) = \kappa f(\tau_i) - \frac{h^2}{24} f''(\tau_i), \quad i = 1, \ldots, N. \tag{3.10}$$

Now using (2.7), (3.10) leads to the matrix-vector equation

$$(A^D + c\kappa B^D)\mathbf{u} = \mathbf{f}, \tag{3.11}$$

where

$$\mathbf{u} = [u_1, u_2, \ldots, u_N]^T, \quad \mathbf{f} = \left[\kappa f(\tau_1) - \frac{h^2}{24} f''(\tau_1), \ldots, \kappa f(\tau_N) - \frac{h^2}{24} f''(\tau_N)\right]^T,$$

$$A^D = \frac{1}{h^2} \begin{bmatrix} -3 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -3 \end{bmatrix}_{N\times N} \tag{3.12}$$

$$B^D = \frac{1}{8} \begin{bmatrix} 5 & 1 & & & \\ 1 & 6 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 6 & 1 \\ & & & 1 & 5 \end{bmatrix}_{N\times N} \tag{3.13}$$

Using (2.11), the matrices A^D and B^D can be written as

$$A^D = \frac{1}{h^2} T, \tag{3.14}$$

$$B^D = \frac{1}{8} (T + 8I). \tag{3.15}$$

where I is the $N \times N$ identity matrix. Substituting into (3.11), we get the system

$$\left[\left(\frac{1}{h^2} + \frac{1}{8} c\kappa\right) T + c\kappa I\right] \mathbf{u} = \mathbf{f}. \tag{3.16}$$

This is a simple tridiagonal system and can be solved easily using a tridiagonal solver at a cost of $O(N)$ operations.

3.2 Existence and Uniqueness

To prove the existence and uniqueness of the solution, we show that the coefficient matrix of (3.11) is nonsingular, which we prove by setting $\mathbf{f} = \mathbf{0}$ and showing that $\mathbf{u} = \mathbf{0}$.

We start by assuming that $\mathbf{u} \neq \mathbf{0}$. Then multiplying (3.11) by $\mathbf{u}^T$ to get

$$\mathbf{u}^T A^D \mathbf{u} + c\kappa \mathbf{u}^T B^D \mathbf{u} = 0. \tag{3.17}$$

From (2.15), (3.14), and (3.15), we know

$$A^D = Z\Lambda_A Z^T, \quad B^D = Z\Lambda_B Z^T, \tag{3.18}$$

where

$$\Lambda_A = \frac{1}{h^2}\Lambda_T, \quad \Lambda_B = \frac{1}{8}(\Lambda_T + 8I), \tag{3.19}$$

where I is the $N \times N$ identity matrix. Note that equations (3.19) and (2.15) imply that both matrices A^D and B^D are nonsingular since neither has a zero eigenvalue. Substituting (3.19) into (3.17), we get

$$\mathbf{u}^T Z\Lambda_A Z^T \mathbf{u} + c\kappa \mathbf{u}^T Z\Lambda_B Z^T \mathbf{u} = 0. \tag{3.20}$$

Then we set

$$\mathbf{q} = Z^T \mathbf{u}. \tag{3.21}$$

Note that since Z^T is nonsingular, $\mathbf{u} \neq \mathbf{0}$ implies $\mathbf{q} \neq \mathbf{0}$. Substituting (3.21) into (3.20), we have

$$\sum_{i=1}^{N} q_i^2 \lambda_i^A + c\kappa \sum_{i=1}^{N} q_i^2 \lambda_i^B = 0. \tag{3.22}$$

Since $c < 0$, we have $c\kappa < 0$. From (3.19) and (2.15), we have

$$\lambda_i^A < 0, \quad \lambda_i^B > 0. \tag{3.23}$$

Hence,

$$\sum_{i=1}^{N} q_i^2 \lambda_i^A + c\kappa \sum_{i=1}^{N} q_i^2 \lambda_i^B < 0, \tag{3.24}$$

which contradicts (3.22). Therefore, $\mathbf{u} = \mathbf{0}$.

3.3 Numerical Results

To demonstrate the accuracy of (3.8) for the solution of (3.1)–(3.2), three test problems are considered. For each test problem, several values of N are chosen. For each value of N, eight quantities for $u(x)$ and $u'(x)$ are given, namely,

1. $E_n(N)$ — the maximum absolute error at the nodes $\{x_i\}_{i=0}^{N}$;

2. $E_c(N)$ — the maximum absolute error at the collocation points $\{\tau_i\}_{i=1}^{N}$;

3. $E_G(N)$ — the maximum absolute error at the Gauss points $\{\eta_i\}_{i=1}^{2N}$, where $\eta_i = x_i + h(3-\sqrt{3})/6$, and $\eta_{i+1} = x_i + h(3+\sqrt{3})/6, \quad i = 1, 3, \ldots, 2N-1$;

4. $E_g(N)$ — an estimate of the global error computed at 10 equally spaced points in each subinterval $I_i, \quad i = 1, 2, \ldots, N$;

5. $R_n(N)$ — the experimental convergence rate of the error at the nodes computed using
$$R_n(N) = \frac{\log[E_n(N/2)] - \log[E_n(N)]}{\log 2}; \tag{3.25}$$

6. $R_c(N)$ — the experimental convergence rate of the error at the collocation points computed using (3.25) with E_c replacing E_n;

7. $R_G(N)$ — the experimental convergence rate of the error at the Gauss points computed using (3.25) with E_G replacing E_n;

8. $R_g(N)$ — the experimental convergence rate of the estimated global error computed using (3.25) with E_g replacing E_n.

The value of the $u_h''(x)$ is not defined at the nodes, thus maximum errors and convergence rates are not computed at the nodal points. The corresponding columns are left blank in all tables.

Problem 3.1:

$$c = -9, \quad u(x) = \cosh\left(3x - \frac{3}{2}\right) - \cosh\left(\frac{3}{2}\right). \tag{3.26}$$

Problem 3.2:

$$c = -\pi^2, \quad u(x) = 1 - \cos(2\pi x). \tag{3.27}$$

Problem 3.3:

$$c = -9, \quad u(x) = e^x(x^2 - x).$$

The results for Problems 3.1–3.3 are given in Tables 3.1–3.3, respectively. From the results, we see that the method is globally optimal, that is, third-order accurate for u, second-order for accurate u', and first-order accurate for u''. The method also exhibits superconvergence results: fourth-order accuracy of the u approximation at the nodal and collocation points, third-order accuracy of the u' approximation at the Gauss points, and second-order accuracy of the u'' approximation at the collocation points. Problem 3.1 is used in [11]. Comparatively for u and a given value of N, our error is globally smaller by a factor of 10^{-2} than the error reported in [11] Table 2.1. The convergence rates are comparable.

Table 3.1: Test Problem 3.1.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	3.2185(-5)	2.9376(-4)	9.5125(-4)	9.3631(-4)				
16	2.0324(-6)	2.0465(-5)	1.1670(-4)	1.1372(-4)	3.9852	3.8434	3.0270	3.0415
32	1.2735(-7)	1.3483(-6)	1.4360(-5)	1.4055(-5)	3.9963	3.9239	3.0227	3.0163
64	7.9645(-9)	8.6483(-8)	1.7778(-6)	1.7443(-6)	3.9991	3.9626	3.0139	3.0103
128	4.9785(-10)	5.4751(-9)	2.2105(-7)	2.1718(-7)	3.9998	3.9814	3.0076	3.0057
256	3.1108(-11)	3.4439(-10)	2.7556(-8)	2.7091(-8)	4.0003	3.9908	3.0040	3.0030
u'								
8	7.3629(-2)	3.0487(-2)	5.1288(-3)	3.9301(-2)				
16	1.8636(-2)	8.4421(-3)	6.9491(-4)	9.7039(-3)	1.9821	1.8525	2.8837	2.0179
32	4.6737(-3)	2.2216(-3)	9.0121(-5)	2.3971(-3)	1.9955	1.9260	2.9469	2.0173
64	1.1693(-3)	5.6990(-4)	1.1463(-5)	5.9477(-4)	1.9989	1.9628	2.9748	2.0109
128	2.9239(-4)	1.4433(-4)	1.4451(-6)	1.4807(-4)	1.9997	1.9814	2.9878	2.0060
256	7.3102(-5)	3.6316(-5)	1.8139(-7)	3.6937(-5)	1.9999	1.9907	2.9940	2.0032
u''								
8		1.0243(-1)	1.8897	2.7023				
16		2.8330(-2)	9.9153(-1)	1.4108		1.8543	0.9304	0.9376
32		7.4206(-3)	5.0724(-1)	7.2022(-1)		1.9327	0.9670	0.9701
64		1.8970(-3)	2.5647(-1)	3.6380(-1)		1.9679	0.9839	0.9853
128		4.7943(-4)	1.2895(-1)	1.8282(-1)		1.9843	0.9920	0.9927
256		1.2050(-4)	6.4651(-2)	9.1640(-2)		1.9923	0.9960	0.9964

Table 3.2: Test Problem 3.2.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	3.2427(-3)	2.6919(-3)	5.3373(-3)	5.2419(-3)				
16	1.9702(-4)	1.7526(-4)	5.6692(-4)	5.5808(-4)	4.0408	3.9411	3.2349	3.2315
32	1.2227(-5)	1.1252(-5)	6.5531(-5)	6.4518(-5)	4.0102	3.9612	3.1129	3.1127
64	7.6282(-7)	7.1400(-7)	7.8823(-6)	7.7584(-6)	4.0026	3.9781	3.0555	3.0559
128	4.7656(-8)	4.4984(-8)	9.6665(-7)	9.5125(-7)	4.0006	3.9884	3.0275	3.0279
256	2.9780(-9)	2.8231(-9)	1.1969(-7)	1.1777(-7)	4.0002	3.9941	3.0137	3.0139
u'								
8	3.3135(-1)	1.5935(-1)	4.9787(-2)	1.7146(-1)				
16	8.1266(-2)	4.0252(-2)	6.1531(-3)	4.1233(-2)	2.0276	1.9850	3.0164	2.0560
32	2.0219(-2)	1.0086(-2)	7.7681(-4)	1.0209(-2)	2.0070	1.9968	2.9857	2.0139
64	5.0486(-3)	2.5228(-3)	9.6570(-5)	2.5462(-3)	2.0017	1.9992	3.0079	2.0035
128	1.2618(-3)	6.3079(-4)	1.2006(-5)	6.3616(-4)	2.0004	1.9998	3.0078	2.0009
256	3.1542(-4)	1.5770(-4)	1.4956(-6)	1.5902(-4)	2.0001	1.9999	3.0049	2.0002
u''								
8		9.3934(-1)	8.9813	1.2687				
16		2.4889(-1)	4.4795	6.3429		1.9161	1.0036	1.0002
32		6.3118(-2)	2.2382	3.1712		1.9794	1.0010	1.0001
64		1.5836(-2)	1.1189	1.5856		1.9949	1.0002	1.0000
128		3.9624(-3)	5.5943(-1)	7.9277(-1)		1.9987	1.0001	1.0000
256		9.9082(-4)	2.7971(-1)	3.9639(-1)		1.9997	1.0000	1.0000

Table 3.3: Test Problem 3.3.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	1.8965(-5)	6.8097(-5)	3.9527(-4)	3.8525(-4)				
16	1.1879(-6)	4.7006(-6)	4.8858(-5)	4.7816(-5)	3.9969	3.8567	3.0162	3.0102
32	7.4283(-8)	3.0860(-7)	6.0537(-6)	5.9391(-6)	3.9992	3.9290	3.0127	3.0092
64	4.6433(-9)	1.9765(-8)	7.5274(-7)	7.3949(-7)	3.9998	3.9647	3.0076	3.0057
128	2.9022(-10)	1.2505(-9)	9.3825(-8)	9.2238(-8)	4.0000	3.9824	3.0041	3.0031
256	1.8145(-11)	7.8632(-11)	1.1711(-8)	1.1517(-8)	3.9995	3.9912	3.0021	3.0016
u'								
8	3.1591(-2)	1.4344(-2)	1.3115(-3)	1.6435(-2)				
16	7.9471(-3)	3.7734(-3)	1.6854(-4)	4.0775(-3)	1.9910	1.9265	2.9601	2.0110
32	1.9899(-3)	9.6865(-4)	2.1228(-5)	1.0125(-3)	1.9977	1.9618	2.9891	2.0097
64	4.9767(-4)	2.4546(-4)	2.6589(-6)	2.5209(-4)	1.9994	1.9805	2.9971	2.0060
128	1.2443(-4)	6.1788(-5)	3.3254(-7)	6.2881(-5)	1.9999	1.9901	2.9992	2.0033
256	3.1108(-5)	1.5500(-5)	4.1573(-8)	1.5702(-5)	2.0000	1.9950	2.9998	2.0017
u''								
8		2.5062(-2)	8.4057(-1)	1.1966				
16		6.6986(-3)	4.3097(-1)	6.1202(-1)		1.9036	0.9638	0.9673
32		1.7242(-3)	2.1810(-1)	3.0939(-1)		1.9579	0.9826	0.9842
64		4.3688(-4)	1.0970(-1)	1.5554(-1)		1.9806	0.9914	0.9922
128		1.0992(-4)	5.5012(-2)	7.7979(-2)		1.9907	0.9957	0.9961
256		2.7567(-5)	2.7547(-2)	3.9042(-2)		1.9955	0.9979	0.9981

Chapter 4

POISSON'S EQUATION

This chapter generalizes the method developed in Chapter 3 to approximate the solution of (1.1).

4.1 Derivation

To generalize the method of Chapter 3, we begin by considering properties of the interpolant introduced by Christara [13], which we denote by $w_h(x, y)$ and is defined by

$$\begin{cases} w_h(\tau_i, \tau_j) = u(\tau_i, \tau_j), & i, j = 1, \ldots, N, \\ w_h(\alpha, \tau_j) = u(\alpha, \tau_j) - \frac{h^4}{128} D_x^4 u(\alpha, \tau_j), & \alpha = 0, 1; \quad j = 1, \ldots, N, \\ w_h(\tau_i, \beta) = u(\tau_i, \beta) - \frac{h^4}{128} D_y^4 u(\tau_i, \beta), & j = 1, \ldots, N; \quad \beta = 0, 1, \end{cases} \tag{4.1}$$

and, at each of the corners of Ω, w_h satisfies one of the relations

$$\begin{cases} w_h(\alpha, \beta) = u(\alpha, \beta) - \frac{h^4}{128} D_x^4 u(\alpha, \beta), \\ w_h(\alpha, \beta) = u(\alpha, \beta) - \frac{h^4}{128} D_y^4 u(\alpha, \beta), \end{cases} \tag{4.2}$$

where $\alpha, \beta = 0, 1$. Then from [13, (2.6)], we have

$$D_x^2 w_h(\tau_i, \tau_j) = D_x^2 u(\tau_i, \tau_j) - \frac{h^2}{24} D_x^4 u(\tau_i, \tau_j) + O(h^4), \tag{4.3}$$

and

$$D_y^2 w_h(\tau_i, \tau_j) = D_y^2 u(\tau_i, \tau_j) - \frac{h^2}{24} D_y^4 u(\tau_i, \tau_j) + O(h^4). \tag{4.4}$$

Now we prove that

$$D_x^2 D_y^2 w_h(\tau_i, \tau_j) = D_x^2 D_y^2 u(\tau_i, \tau_j) + O(h^2). \tag{4.5}$$

To begin, we denote by S_x, the one-dimensional quadratic spline interpolation operator in the x-direction

$$S_x : C([0,1]) \to S_2, \tag{4.6}$$

defined in [13] by the interpolation condition

$$\begin{cases} (S_x u)(\tau_i) = u(\tau_i), & i = 1, \ldots, N, \\ (S_x u)(\alpha) = u(\alpha) - \dfrac{h^4}{128} D_x^4 u(\alpha), & \alpha = 0, 1, \end{cases} \tag{4.7}$$

and with S_y defined similarly. From [13, Lemma 2.1], we have

$$S_{xy} = S_x S_y, \tag{4.8}$$

where S_{xy} denotes the two-dimensional quadratic spline interpolation operator defined in [13] by (4.1) and (4.2). Note that $w_h = S_{xy} u$. Then, on using (4.8),

$$\begin{aligned} D_x^2 D_y^2 w_h - D_x^2 D_y^2 u &= D_x^2 D_y^2 S_{xy} u - D_x^2 D_y^2 u \\ &= D_x^2 S_x D_y^2 S_y u - D_x^2 D_y^2 u \\ &= D_x^2 S_x (D_y^2 S_y u - D_y^2 u) + D_x^2 S_x (D_y^2) - D_x^2 (D_y^2 u). \end{aligned}$$

Now setting $z = D_y^2 S_y u - D_y^2 u$, we have

$$\begin{aligned} D_x^2 D_y^2 S_{xy} u(\tau_i, \tau_j) - D_x^2 D_y^2 u(\tau_i, \tau_j) &= D_x^2 S_x z(\tau_i, \tau_j) - D_x^2 z(\tau_i, \tau_j) + D_x^2 (D_y^2 S_y u(\tau_i, \tau_j) \\ &\quad - D_y^2 u(\tau_i, \tau_j)) + D_x^2 S_x (D_y^2 u(\tau_i, \tau_j)) - D_x^2 (D_y^2 u(\tau_i, \tau_j)). \end{aligned}$$

Then upon using (4.3) and (4.4),

$$D_x^2 D_y^2 S_{xy} u(\tau_i, \tau_j) - D_x^2 D_y^2 u(\tau_i, \tau_j) = \frac{h^4}{576} D_x^4 D_y^4 u(\tau_i, \tau_j) + O(h^6) - \frac{h^2}{24} D_x^2 D_y^4 u(\tau_i, \tau_j)$$
$$+ O(h^4) - \frac{h^2}{24} D_x^4 D_y^2 u(\tau_i, \tau_j) + O(h^4)$$
$$= O(h^2),$$

as desired.

Adding (4.3) and (4.4), we have

$$\Delta w_h(\tau_i, \tau_j) = D_x^2 u(\tau_i, \tau_j) + D_y^2 u(\tau_i, \tau_j) - \frac{h^2}{24}[D_x^4 u(\tau_i, \tau_j) + D_y^4 u(\tau_i, \tau_j)] + O(h^4)$$
$$= \Delta u(\tau_i, \tau_j) - \frac{h^2}{24}[\Delta^2 u(\tau_i, \tau_j) - 2D_x^2 D_y^2 u(\tau_i, \tau_j)] + O(h^4),$$

and on using (1.1),

$$\Delta w_h(\tau_i, \tau_j) = f(\tau_i, \tau_j) - \frac{h^2}{24}\Delta f(\tau_i, \tau_j) + \frac{h^2}{12} D_x^2 D_y^2 u(\tau_i, \tau_j) + O(h^4). \tag{4.9}$$

Then using (4.5), we obtain

$$\Delta w_h(\tau_i, \tau_j) - \frac{h^2}{12} D_x^2 D_y^2 w_h(\tau_i, \tau_j) = f(\tau_i, \tau_j) - \frac{h^2}{24}\Delta f(\tau_i, \tau_j) + O(h^4). \tag{4.10}$$

Finally, replacing w_h with u_h and dropping the $O(h^4)$ term, it follows that

$$\Delta u_h(\tau_i, \tau_j) - \frac{h^2}{12} D_x^2 D_y^2 u_h(\tau_i, \tau_j) = f(\tau_i, \tau_j) - \frac{h^2}{24}\Delta f(\tau_i, \tau_j), \quad i, j = 1, \ldots, N. \tag{4.11}$$

We then seek $u_h \in S_2^D \otimes S_2^D$ such that (4.11) is satisfied. Taking

$$u_h(x, y) = \sum_{m=1}^{N} \sum_{n=1}^{N} u_{m,n} \mathcal{B}_m^D(x) \mathcal{B}_n^D(y), \tag{4.12}$$

and substituting into (4.11), we get

$$\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=1}^{N}\mathcal{B}_n(\tau_j)u_{m,n}+\sum_{m=1}^{N}\mathcal{B}_m^D(\tau_i)\sum_{n=1}^{N}[\mathcal{B}_n^D(\tau_j)]''u_{m,n}$$
$$-\frac{h^2}{12}\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=1}^{N}[\mathcal{B}_n^D(\tau_j)]''u_{m,n}=f(\tau_i,\tau_j)-\frac{h^2}{24}\Delta f(\tau_i,\tau_j). \tag{4.13}$$

Then using (2.7), (2.17), and (2.18), we obtain the linear system

$$\left[A^D\otimes B^D+\left(B^D-\frac{h^2}{12}A^D\right)\otimes A^D\right]\mathbf{u}=\mathbf{f}, \tag{4.14}$$

where A^D and B^D are given in (3.12) and (3.13), respectively,

$$\mathbf{u}=[u_{1,1},\ldots,u_{1,N},\ldots,u_{N,1},\ldots,u_{N,N}]^T,$$

and

$$\mathbf{f}=[f_{1,1},\ldots,f_{1,N},\ldots,f_{N,1},\ldots,f_{N,N}]^T,\quad f_{i,j}=f(\tau_i,\tau_j)-\frac{h^2}{24}\Delta f(\tau_i,\tau_j).$$

4.2 Solving the System

Substituting (3.14) and (3.15) into (4.14) and simplifying, we obtain

$$\frac{1}{h^2}\left[T\otimes I+\frac{1}{6}(T+6I)\otimes T\right]\mathbf{u}=\mathbf{f}. \tag{4.15}$$

where T is given in (2.11) and I is the $N\times N$ identity matrix. Now, (4.15) is equivalent to

$$(Z^T\otimes I)\left[T\otimes I+\frac{1}{6}(T+6I)\otimes T\right](Z\otimes I)(Z^{-1}\otimes I)\mathbf{u}=h^2(Z^T\otimes I)\mathbf{f}, \tag{4.16}$$

where Z is given in (2.13)–(2.14). Then using (2.15), we have

$$\left[\Lambda_T \otimes I + \frac{1}{6}(\Lambda_T + 6I) \otimes T\right] \widetilde{\mathbf{u}} = h^2 \widetilde{\mathbf{f}}, \tag{4.17}$$

where $\widetilde{\mathbf{u}} = (Z^T \otimes I)\mathbf{u}$ and $\widetilde{\mathbf{f}} = (Z^T \otimes I)\mathbf{f}$. Since Λ_T and $\Lambda_T + 6I$ are diagonal, (4.17) reduces to the independent systems

$$\left[\lambda_i^T I + \left(\frac{1}{6}\lambda_i^T + 1\right) T\right] \widetilde{\mathbf{u}}_\mathbf{i} = h^2 \widetilde{\mathbf{f}}_\mathbf{i}, \quad i = 1, \ldots, N, \tag{4.18}$$

where

$$\widetilde{\mathbf{u}}_\mathbf{i} = [\widetilde{u}_{i,1}, \ldots, \widetilde{u}_{i,N}]^T, \quad \widetilde{\mathbf{f}} = [\widetilde{f}_{i,1}, \ldots, \widetilde{f}_{i,N}]^T.$$

This leads to the following **Matrix Decomposition Algorithm**:

Step 1. Compute

$$\widetilde{\mathbf{f}} = (Z^T \otimes I)\mathbf{f}.$$

Step 2. Solve the N tridiagonal systems

$$\left[\lambda_i^T I + \left(\frac{1}{6}\lambda_i^T + 1\right) T\right] \widetilde{\mathbf{u}}_\mathbf{i} = h^2 \widetilde{\mathbf{f}}_\mathbf{i}, \quad i = 1, \ldots, N.$$

Step 3. Compute

$$\mathbf{u} = (Z \otimes I)\widetilde{\mathbf{u}}.$$

Since the elements of the matrix Z are sines, steps 1 and 3 can employ fast Fourier transforms (FFTs) to perform the matrix multiplication at a cost of $O(N^2 \log N)$ operations. Since the coefficient matrix in step 2 is tridiagonal, the solution can be obtained at a cost of $O(N^2)$ operations. Therefore, the total cost of this algorithm is $O(N^2 \log N)$.

4.3 Existence and Uniqueness

We show the existence and uniqueness of the solution by proving that the coefficient matrix of (4.18) is nonsingular. We do this by setting $\mathbf{f} = \mathbf{0}$ and showing that $\mathbf{u} = \mathbf{0}$.

To begin, we assume that $\mathbf{u} \neq \mathbf{0}$. Then we multiply (4.18) by $\mathbf{u}^T$ to get

$$\lambda_i^T \mathbf{u}^T \mathbf{u} + \left(\frac{1}{6}\lambda_i^T + 1\right) \mathbf{u}^T T \mathbf{u} = 0, \tag{4.19}$$

where $1 \leq i \leq N$. From (2.15) we have

$$\mathbf{u}^T T \mathbf{u} = \mathbf{u}^T Z \Lambda_T Z^T \mathbf{u}. \tag{4.20}$$

Then we set

$$\mathbf{q} = Z^T \mathbf{u}. \tag{4.21}$$

Note that since Z^T is nonsingular, $\mathbf{u} \neq \mathbf{0}$ implies $\mathbf{q} \neq \mathbf{0}$. Substituting (4.21) into (4.20) yields

$$\sum_{i=1}^{N} q_i^2 \lambda_i^T < 0, \tag{4.22}$$

since $-4 \leq \lambda_i^T < 0$. Now, we know that $\lambda_i^T \mathbf{u}^T \mathbf{u} < 0$ and $\left(\frac{1}{6}\lambda_i^T + 1\right) > 0$. Hence,

$$\lambda_i^T \mathbf{u}^T \mathbf{u} + \left(\frac{1}{6}\lambda_i^T + 1\right) \mathbf{u}^T T \mathbf{u} < 0, \tag{4.23}$$

which contradicts (4.19). Therefore, $\mathbf{u} = \mathbf{0}$.

4.4 Numerical Results

Three test problems are used to demonstrate the accuracy of the method. Each test problem has the same eight values described in Chapter 3 for u, u_x, u_y, u_{xy}, u_{xx}, and u_{yy}. Note that for u_{xx} and u_{yy}, the value of the approxmation is not defined at the nodal points, hence there are no values for maximum error or convergence rates. For Problems 4.2 and 4.3, values for u_y and u_{yy} are not given because the values are the same as u_x and u_{xx}, respectively.

Problem 4.1:

$$u(x,y) = [1 - \cos(2\pi x)]\sin(2\pi y).$$

Problem 4.2:

$$u(x,y) = (x^2 - x)(y^2 - y)e^{xy}.$$

Problem 4.3:

$$u(x,y) = (x^2 - x)(y^2 - y)e^{x-y}.$$

Problem 4.2 with u multiplied by 3 was used in [1]. The results for Problem 4.1–4.3 are given in Tables 4.1–4.3, respectively. We observe that (4.11) is globally optimal—third-order accurate for u, second-order accurate for u_x, u_y, u_{xy}, and first-order accurate for u_{xx}, u_{yy}. Furthermore, the method is superconvergent. We observe fourth-order accuracy at the nodal and collocation points of the u approximation, third-order accuracy at the Gauss points of the u_x, u_y, u_{xy} approximations, and second-order accuracy at the collocation points of the u_{xx}, u_{yy} approximations.

Table 4.1: Test Problem 4.1.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	8.9801(-3)	1.6064(-3)	1.0062(-2)	1.0734(-2)				
16	5.5107(-4)	1.3808(-4)	1.0488(-3)	1.0589(-3)	4.0264	3.5402	3.2621	3.3416
32	3.4275(-5)	1.0044(-5)	1.2431(-4)	1.2336(-4)	4.0070	3.7811	3.0767	3.1016
64	2.1396(-6)	6.7560(-7)	1.5275(-5)	1.5066(-5)	4.0018	3.8940	3.0247	3.0335
128	1.3368(-7)	4.3775(-8)	1.9001(-6)	1.8705(-6)	4.0004	3.9480	3.0070	3.0097
u_x								
8	3.0645(-1)	1.5107(-1)	5.2351(-2)	1.7828(-1)				
16	7.9805(-2)	3.9759(-2)	6.6852(-3)	4.0774(-2)	1.9411	1.9259	2.9692	2.1284
32	2.0129(-2)	1.0054(-2)	8.3225(-4)	1.0143(-2)	1.9872	1.9834	3.0059	2.0071
64	5.0430(-3)	2.5208(-3)	1.0079(-4)	2.5421(-3)	1.9969	1.9959	3.0457	1.9965
128	1.2614(-3)	6.3067(-4)	1.2294(-5)	6.3591(-4)	1.9992	1.9989	3.0354	1.9991
u_y								
8	6.2895(-1)	3.1673(-1)	1.0297(-1)	3.4237(-1)				
16	1.6051(-1)	8.0482(-2)	1.2377(-2)	8.0845(-2)	1.9702	1.9765	3.0566	2.0823
32	4.0313(-2)	2.0172(-2)	1.5310(-3)	2.0317(-2)	1.9934	1.9963	3.0151	1.9925
64	1.0089(-2)	5.0457(-3)	1.9087(-4)	5.0860(-3)	1.9984	1.9992	3.0038	1.9981
128	2.5231(-3)	1.2616(-3)	2.3843(-5)	1.2719(-3)	1.9996	1.9998	3.0010	1.9995
u_{xy}								
8	4.1939	1.8842	3.5255(-1)	2.1048				
16	1.0233	4.9881(-1)	4.3419(-2)	5.1506(-1)	2.0351	1.9174	3.0214	2.0309
32	2.5421(-1)	1.2630(-1)	5.2740(-3)	1.2810(-1)	2.0091	1.9816	3.0414	2.0075
64	6.3451(-2)	3.1675(-2)	6.3480(-4)	3.1984(-2)	2.0023	1.9955	3.0545	2.0018
128	1.5856(-2)	7.9251(-3)	7.7293(-5)	7.9935(-3)	2.0006	1.9989	3.0379	2.0005
u_{xx}								
8		8.3752(-1)	8.8193	1.2608(+1)				
16		2.4182(-1)	4.4636	6.3381		1.7922	0.9824	0.9922
32		6.2664(-2)	2.2363	3.1707		1.9482	0.9971	0.9993
64		1.5807(-2)	1.1187	1.5855		1.9871	0.9993	0.9999
128		3.9606(-3)	5.5940(-1)	7.9277(-1)		1.9968	0.9998	1.0000
u_{yy}								
8		1.7452	1.7762(+1)	2.5256(+1)				
16		4.8877(-1)	8.9405	1.2678(+1)		1.8362	0.9903	0.9943
32		1.2566(-1)	4.4743	6.3417		1.9596	0.9987	0.9994
64		3.1635(-2)	2.2376	3.1710		1.9899	0.9997	0.9999
128		7.9226(-3)	1.1188	1.5855		1.9975	0.9999	1.0000

Table 4.2: Test Problem 4.2.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	8.0387(-6)	3.9893(-6)	5.9111(-5)	5.9007(-5)				
16	5.1659(-7)	3.1836(-7)	7.1286(-6)	7.0856(-6)	3.9599	3.6474	3.0517	3.0579
32	3.2318(-8)	2.2648(-8)	8.8470(-7)	8.7394(-7)	3.9986	3.8132	3.0103	3.0193
64	2.0278(-9)	1.5095(-9)	1.0988(-7)	1.0834(-7)	3.9943	3.9072	3.0093	3.0120
128	1.2673(-10)	9.7554(-11)	1.3678(-8)	1.3471(-8)	4.0001	3.9518	3.0060	3.0076
u_x								
8	3.0966(-3)	1.4460(-3)	3.2965(-4)	1.7866(-3)				
16	7.9293(-4)	3.8172(-4)	4.5190(-5)	4.3149(-4)	1.9654	1.9215	2.8669	2.0498
32	1.9946(-4)	9.7742(-5)	5.8853(-6)	1.0477(-4)	1.9911	1.9655	2.9408	2.0421
64	4.9943(-5)	2.4717(-5)	7.6538(-7)	2.5720(-5)	1.9978	1.9834	2.9429	2.0262
128	1.2491(-5)	6.2131(-6)	9.7703(-8)	6.3657(-6)	1.9994	1.9921	2.9697	2.0145
u_{xy}								
8	6.0757(-2)	1.8954(-2)	1.0791(-3)	2.9018(-2)				
16	1.5687(-2)	6.1571(-3)	2.4148(-4)	7.7120(-3)	1.9535	1.6221	2.1598	1.9118
32	3.9633(-3)	1.7500(-3)	3.7963(-5)	1.9751(-3)	1.9848	1.8149	2.6692	1.9652
64	9.9409(-4)	4.6647(-4)	5.1875(-6)	4.9847(-4)	1.9953	1.9075	2.8715	1.9863
128	2.4877(-4)	1.2044(-4)	6.6840(-7)	1.2509(-4)	1.9986	1.9535	2.9563	1.9945
u_{xx}								
8		1.7586(-3)	8.5555(-2)	1.2156(-1)				
16		5.1618(-4)	4.3558(-2)	6.1807(-2)		1.7685	0.9739	0.9758
32		1.3644(-4)	2.1963(-2)	3.1147(-2)		1.9196	0.9879	0.9887
64		3.4762(-5)	1.1029(-2)	1.5635(-2)		1.9727	0.9938	0.9943
128		8.7586(-6)	5.5265(-3)	7.8334(-3)		1.9887	0.9968	0.9971

Table 4.3: Test Problem 4.3.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	1.0650(-5)	1.0110(-5)	7.2999(-5)	7.3473(-5)				
16	6.8156(-7)	7.2852(-7)	9.0305(-6)	8.9816(-6)	3.9658	3.7946	3.0150	3.0322
32	4.2602(-8)	4.8784(-8)	1.1189(-6)	1.1067(-6)	3.9998	3.9005	3.0127	3.0207
64	2.6632(-9)	3.1544(-9)	1.3916(-7)	1.3728(-7)	3.9997	3.9510	3.0073	3.0111
128	1.6652(-10)	2.0050(-10)	1.7354(-8)	1.7096(-8)	3.9994	3.9757	3.0034	3.0054
u_x								
8	5.0280(-3)	2.2877(-3)	3.6669(-4)	2.7729(-3)				
16	1.2763(-3)	6.0706(-4)	5.1244(-5)	6.7652(-4)	1.9780	1.9140	2.8391	2.0352
32	3.2030(-4)	1.5607(-4)	6.6986(-6)	1.6583(-4)	1.9945	1.9596	2.9355	2.0284
64	8.0151(-5)	3.9553(-5)	8.5375(-7)	4.0965(-5)	1.9986	1.9803	2.9720	2.0172
128	2.0047(-5)	9.9551(-6)	1.0769(-7)	1.0175(-5)	1.9993	1.9903	2.9870	2.0093
u_{xy}								
8	6.1994(-2)	2.2336(-2)	1.8096(-3)	3.0605(-2)				
16	1.5794(-2)	6.6741(-3)	2.9378(-4)	7.8975(-3)	1.9728	1.7427	2.6229	1.9543
32	3.9719(-3)	1.8220(-3)	4.0755(-5)	1.9960(-3)	1.9915	1.8730	2.8497	1.9843
64	9.9474(-4)	4.7604(-4)	5.2934(-6)	5.0084(-4)	1.9974	1.9364	2.9447	1.9947
128	2.4881(-4)	1.2168(-4)	6.6933(-7)	1.2537(-4)	1.9992	1.9680	2.9834	1.9982
u_{xx}								
8		3.8909(-3)	1.3563(-1)	1.9299(-1)				
16		1.0691(-3)	6.9497(-2)	9.8670(-2)		1.8637	0.9646	0.9679
32		2.7720(-4)	3.5151(-2)	4.9859(-2)		1.9474	0.9834	0.9848
64		7.0356(-5)	1.7675(-2)	2.5061(-2)		1.9782	0.9918	0.9924
128		1.7708(-5)	8.8635(-3)	1.2564(-2)		1.9903	0.9958	0.9962

Chapter 5

FOURTH-ORDER TWO-POINT BOUNDARY VALUE PROBLEM

This chapter deals with the solution of the problem

$$u^{(4)}(x) = f(x), \quad x \in [0,1], \tag{5.1}$$

$$u(0) = u(1) = u'(0) = u'(1) = 0. \tag{5.2}$$

The developed method is generalized in Chapter 6 to solve the auxiliary biharmonic problem.

5.1 Derivation

First, we write (5.1) as the coupled system of equations,

$$u''(x) = v(x), \tag{5.3}$$

$$v''(x) = f(x), \tag{5.4}$$

subject to the boundary conditions (5.2). Then we observe that (5.3) is the same as (3.1) with $c = 0$ and v replacing f, and (5.4) is the same as (3.1) with $c = 0$ and v replacing u. Thus, to solve (5.3) and (5.4) with the boundary conditions (5.2), we apply (3.8), and seek $u_h \in S_2^D$ and $v_h \in S_2$ such that

$$u_h''(\tau_i) - v_h(\tau_i) = -\frac{h^2}{24} f(\tau_i), \quad i = 1, \ldots, N, \tag{5.5}$$

$$v_h''(\tau_i) = f(\tau_i) - \frac{h^2}{24} f''(\tau_i), \quad i = 1, \ldots, N. \tag{5.6}$$

Now, to obtain an optimal method, we perturb the Neumann boundary conditions. Noting that $u^{(3)} = v'$, equation (2.10) yields

$$u'(0) = w_h'(0) + \frac{h^2}{12}v'(0), \quad u'(1) = w_h'(1) + \frac{h^2}{12}v'(1).$$

Replacing w_h and v with u_h and v_h, respectively, and using the Neumann conditions in (5.2) gives the perturbed Neumann boundary conditions

$$u_h'(0) + \frac{h^2}{12}v_h'(0) = 0, \qquad u_h'(1) + \frac{h^2}{12}v_h'(1) = 0. \tag{5.7}$$

Taking

$$u_h(x) = \sum_{m=1}^{N} u_m \mathcal{B}_m^D(x), \tag{5.8}$$

$$v_h(x) = \sum_{m=0}^{N+1} v_m \mathcal{B}_m(x), \tag{5.9}$$

and substituting into (5.5)–(5.7), and upon using (2.5) and (2.7), we get the linear system of equations

$$\begin{cases} A^D\mathbf{u} - B\mathbf{v} = \mathbf{f}_1, \\ A\mathbf{v} = \mathbf{f}_2, \\ C\mathbf{u} + \dfrac{h^2}{12}D\mathbf{v} = \mathbf{0}, \end{cases} \tag{5.10}$$

where A^D is given in (3.12), and

$$\mathbf{u} = [u_1, \ldots, u_N]^T, \quad \mathbf{v} = [v_0, \ldots, v_{N+1}]^T,$$

$$\mathbf{f}_1 = -\frac{h^2}{24}\left[f(\tau_1), \ldots, f(\tau_N)\right]^T, \quad \mathbf{f}_2 = \left[f(\tau_1) - \frac{h^2}{24}f''(\tau_1), \ldots, f(\tau_N) - \frac{h^2}{24}f''(\tau_N)\right]^T,$$

$$A = \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix}_{N\times(N+2)}, \tag{5.11}$$

$$B = \frac{1}{8}\begin{bmatrix} 1 & 6 & 1 & & & \\ & 1 & 6 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 6 & 1 \end{bmatrix}_{N\times(N+2)} \tag{5.12}$$

$$C = \frac{1}{h}\begin{bmatrix} 2 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & -2 \end{bmatrix}_{2\times N}, \tag{5.13}$$

$$D = \frac{1}{h}\begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & -1 \end{bmatrix}_{2\times(N+2)} \tag{5.14}$$

5.2 Existence and Uniqueness

To show that the solution of (5.5)–(5.7) exists and is unique, we examine a more general problem that arises in Chapter 6. Since the coefficient matrix is square, to show existence and uniqueness, we simply show that $f(x) = 0$ implies $u_h(x) = v_h(x) = 0$. To that end, consider the problem

$$u_h(\alpha) = u_h'(\alpha) + \frac{h^2}{12}v_h'(\alpha) = 0, \qquad \alpha = 0, 1, \tag{5.15}$$

$$\frac{1}{h^2}\lambda u_h(\tau_i) + \left(\frac{1}{24}\lambda + 1\right) u_h''(\tau_i) - \left(\frac{1}{8}\lambda + 1\right) v_h(\tau_i) = 0, \tag{5.16}$$

$$\frac{1}{h^2}\lambda v_h(\tau_i) + \left(\frac{1}{24}\lambda + 1\right) v_h''(\tau_i) = 0, \tag{5.17}$$

$$i = 1, \ldots, N,$$

where we assume that

$$-4 \leq \lambda \leq 0. \tag{5.18}$$

Note that (5.7), (5.5), and (5.6) are equivalent to (5.15), (5.16), and (5.17), respectively, with $\lambda = 0$. Now, we take

$$u_h(x) = \sum_{m=1}^{N} u_m \mathcal{B}_m^D(x), \tag{5.19}$$

$$v_h(x) = \sum_{m=0}^{N+1} v_m \mathcal{B}_m(x). \tag{5.20}$$

Note that $u_h \in S_2^D$ satisfies the Dirichlet boundary conditions. Substituting (5.19) and (5.20) into the Neumann boundary conditions and applying (2.5) and (2.7), yields

$$\frac{2}{h}u_1 + \frac{h}{12}(v_1 - v_0) = 0,$$

$$-\frac{2}{h}u_N + \frac{h}{12}(v_N - v_{N+1}) = 0.$$

After rearranging, we get

$$u_1 = \frac{h^2}{24}(v_0 - v_1), \tag{5.21}$$

$$u_N = \frac{h^2}{24}(v_{N+1} - v_N). \tag{5.22}$$

Now, substituting (5.20) into (5.17) and using (2.5), gives

$$\frac{\lambda}{8h^2}[v_{i-1} + 6v_i + v_{i+1}] + \left(\frac{1}{3}\lambda + 8\right)\frac{1}{8h^2}[v_{i-1} - 2v_i + v_{i+1}] = 0. \tag{5.23}$$

Then collecting like terms and simplifying, we have

$$a_1 v_{i-1} + a_2 v_i + a_1 v_{i+1} = 0, \quad i = 1, \ldots, N, \tag{5.24}$$

where

$$a_1 = \lambda + 6, \tag{5.25}$$

$$a_2 = 4\lambda - 12. \tag{5.26}$$

Now, taking $i = 1, N$ and rearranging, (5.24) gives

$$v_0 = -\frac{a_2}{a_1}v_1 - v_2, \tag{5.27}$$

$$v_{N+1} = -\frac{a_2}{a_1}v_N - v_{N-1}. \tag{5.28}$$

Now, using (2.5), we have

$$\left(\frac{1}{8}\lambda + 1\right) v_h(\tau_i) = \left(\frac{1}{8}\lambda + 1\right) \frac{1}{8}[v_{i-1} + 6v_i + v_{i+1}]. \tag{5.29}$$

Noting that (5.24) implies

$$(v_{i-1} + v_{i+1}) = -\frac{a_2}{a_1}v_i,$$

and substituting into (5.29) and simplifying, yields

$$\left(\frac{1}{8}\lambda + 1\right) v_h(\tau_i) = \frac{1}{32}b_1 v_i, \quad i = 1, \ldots, N, \tag{5.30}$$

where

$$b_1 = \frac{(\lambda + 8)(\lambda + 24)}{\lambda + 6}. \tag{5.31}$$

Now, substituting (5.30) into (5.16) and multipying by 6, we get

$$\frac{1}{h^2}[a_1 u_{i-1} + a_2 u_i + a_1 u_{i+1}] - \frac{3}{16}b_1 v_i = 0, \quad i = 2, \ldots, N-1, \tag{5.32}$$

$$\frac{1}{h^2}[a_3 u_1 + a_1 u_2] - \frac{3}{16}b_1 v_1 = 0, \quad i = 1, \tag{5.33}$$

$$\frac{1}{h^2}[a_1 u_{N-1} + a_3 u_N] - \frac{3}{16}b_1 v_N = 0, \quad i = N, \tag{5.34}$$

where

$$a_3 = 3\lambda - 18. \tag{5.35}$$

Substituting (5.21) and then (5.27) into (5.33) and collecting like terms gives

$$\frac{a_1}{h^2}u_2 - \frac{b_4}{16}v_1 - \frac{b_2}{8}v_2 = 0, \tag{5.36}$$

where

$$b_2 = \lambda - 6, \tag{5.37}$$

$$b_4 = \frac{(13\lambda^2 + 24\lambda + 648)}{\lambda + 6}. \tag{5.38}$$

Similarly, using (5.22) and (5.28), equation (5.34) becomes

$$\frac{a_1}{h^2}u_{N-1} - \frac{b_4}{16}v_{N-1} - \frac{b_2}{8}v_N = 0. \tag{5.39}$$

Now, for $i = 2$, (5.32) becomes

$$\frac{1}{h^2}[a_2u_2 + a_1u_3] - \frac{b_3}{8}v_1 - \frac{b_5}{16}v_2 = 0, \tag{5.40}$$

where

$$b_3 = \frac{5\lambda - 6}{3}, \tag{5.41}$$

$$b_5 = \frac{11\lambda^2 + 312\lambda + 1800}{3(\lambda + 6)}, \tag{5.42}$$

after using (5.21) and (5.27) and simplifying. Similarly, substituting (5.22) and (5.28) into

(5.32) for $i = N - 1$ and collecting like terms, yields

$$\frac{1}{h^2}[a_1 u_{N-2} + a_2 u_{N-1}] - \frac{b_5}{16} v_{N-1} - \frac{b_3}{8} v_N = 0. \tag{5.43}$$

Clearly, (5.24) is equivalent to

$$\frac{1}{h^2}[a_1 v_{i-1} + a_2 v_i + a_1 v_{i+1}] = 0, \qquad i = 1, \ldots, N, \tag{5.44}$$

Now, setting

$$\mathbf{u} = [u_2, \ldots, u_{N-1}], \qquad \mathbf{v} = [v_1, \ldots, v_N],$$

equations (5.32), (5.36), (5.39), (5.40), (5.43), and (5.44) can be written as the linear system

$$A_u \mathbf{u} - B_v \mathbf{v} = \mathbf{0}, \tag{5.45}$$

$$A_u^T \mathbf{v} = \mathbf{0}, \tag{5.46}$$

where A_u and B_v are, respectively, the $N \times (N-2)$ and $N \times N$ matrices

$$A_u = \frac{1}{h^2}\begin{bmatrix} a_1 & & & & \\ a_2 & a_1 & & & \\ a_1 & a_2 & a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_1 & a_2 & a_1 \\ & & & a_1 & a_2 \\ & & & & a_1 \end{bmatrix}, \quad B_v = \frac{1}{16}\begin{bmatrix} b_4 & 2b_2 & & & & & \\ 2b_3 & b_5 & & & & & \\ & & 3b_1 & & & & \\ & & & \ddots & & & \\ & & & & 3b_1 & & \\ & & & & & b_5 & 2b_3 \\ & & & & & 2b_2 & b_4 \end{bmatrix}. \tag{5.47}$$

Then multiplying (5.45) by $\mathbf{v}^T$ and (5.46) by $\mathbf{u}^T$, gives

$$\mathbf{v}^T A_u \mathbf{u} - \mathbf{v}^T B_v \mathbf{v} = 0, \tag{5.48}$$

$$\mathbf{u}^T A_u^T \mathbf{v} = 0. \tag{5.49}$$

Since

$$0 = \mathbf{u}^T A_u^T \mathbf{v} = (\mathbf{v}^T A_u \mathbf{u})^T = \mathbf{v}^T A_u \mathbf{u},$$

(5.48) becomes

$$\mathbf{v}^T B_v \mathbf{v} = 0. \tag{5.50}$$

Using (5.47), we have

$$2(b_2 + b_3)v_1 v_2 + b_4 v_1^2 + b_5 v_2^2 + 3b_1 \sum_{i=3}^{N-2} v_i^2 + b_5 v_{N-1}^2 + b_4 v_N^2 + 2(b_2 + b_3)v_{N-1}v_N = 0. \tag{5.51}$$

As $-4 \le \lambda \le 0$, we have

$$\begin{aligned} 32 \le b_1 \le 40, \quad -10 \le b_2 \le -6, \quad -9 \le b_3 \le -2, \\ 108 \le b_4 \le 380, \quad 99 \le b_5 \le 122. \end{aligned} \tag{5.52}$$

Hence,

$$2|(b_2 + b_3)v_1 v_2| \le 38(v_1^2 + v_2^2),$$

which implies

$$2(b_2 + b_3)v_1 v_2 \ge -38(v_1^2 + v_2^2).$$

Similarly,

$$2(b_2 + b_3)v_{N-1} v_N \ge -38(v_{N-1}^2 + v_N^2).$$

So (5.51) becomes

$$0 \geq (b_4 - 38)v_1^2 + (b_5 - 38)v_2^2 + 3b_1 \sum_{i=3}^{N-2} v_i^2 + (b_5 - 38)v_{N-1}^2 + (b_4 - 38)v_N^2. \tag{5.53}$$

By (5.52), all terms are positive. Hence,

$$\mathbf{v} = \mathbf{0}.$$

Substituting into (5.45) and using forward substitution yields

$$\mathbf{u} = \mathbf{0}.$$

From (5.21), (5.22), (5.27), and (5.28), we have

$$u_1 = 0, \quad u_N = 0, \quad v_0 = 0, \quad v_{N+1} = 0.$$

Therefore,

$$u_h(x) = v_h(x) = 0.$$

Note that the existence and uniqueness implies that the coefficient matrix corresponding to the problem (5.15)–(5.17) is nonsingular.

5.3 Solving the System

As written in (5.10), the coefficient matrix is

$$S = \begin{bmatrix} A^D & -B \\ 0 & A \\ C & \frac{h^2}{12}D \end{bmatrix}_{(2N+2)\times(2N+2)} \tag{5.54}$$

If we rewrite the vector of unknowns as

$$\mathbf{w} = [v_0, u_1, v_1, \ldots, u_N, v_N, v_{N+1}]^T,$$

the coefficient matrix becomes

$$\widetilde{S} = \begin{bmatrix} \widetilde{c}_1 & \widetilde{c}_2 & -\widetilde{c}_1 & 0 & & & & & & & & \\ \widetilde{b}_1 & \widetilde{a}_3 & \widetilde{b}_2 & \widetilde{a}_1 & \widetilde{b}_1 & & & & & & & \\ \widetilde{a}_1 & 0 & \widetilde{a}_2 & 0 & \widetilde{a}_1 & 0 & & & & & & \\ & \widetilde{a}_1 & \widetilde{b}_1 & \widetilde{a}_2 & \widetilde{b}_2 & \widetilde{a}_1 & \widetilde{b}_1 & & & & & \\ & & \widetilde{a}_1 & 0 & \widetilde{a}_2 & 0 & \widetilde{a}_1 & 0 & & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & \widetilde{a}_1 & \widetilde{b}_1 & \widetilde{a}_2 & \widetilde{b}_2 & \widetilde{a}_1 & \widetilde{b}_1 & & \\ & & & & & \widetilde{a}_1 & 0 & \widetilde{a}_2 & 0 & \widetilde{a}_1 & 0 & \\ & & & & & & \widetilde{a}_1 & \widetilde{b}_1 & \widetilde{a}_3 & \widetilde{b}_2 & \widetilde{b}_1 & \\ & & & & & & & \widetilde{a}_1 & 0 & \widetilde{a}_2 & \widetilde{a}_1 & \\ & & & & & & & & \widetilde{c}_2 & \widetilde{c}_1 & -\widetilde{c}_1 & \end{bmatrix}, \tag{5.55}$$

where

$$\widetilde{a}_1 = \frac{1}{h^2}, \quad \widetilde{a}_2 = -\frac{2}{h^2}, \quad \widetilde{a}_3 = \frac{3}{h^2}, \quad \widetilde{b}_1 = -\frac{1}{8}, \quad \widetilde{b}_2 = -\frac{3}{4}, \quad \widetilde{c}_1 = \frac{h}{12}, \quad \widetilde{c}_2 = \frac{2}{h}.$$

Thus we have the system of linear equations

$$\widetilde{S}\mathbf{w} = \mathbf{r}, \tag{5.56}$$

where

$$\mathbf{r} = \left[0, -\frac{h^2}{24}f(\tau_1), f(\tau_1) - \frac{h^2}{24}f''(\tau_1), \ldots, \frac{h^2}{24}f(\tau_N), f(\tau_N) - \frac{h^2}{24}f''(\tau_N), 0\right]^T, \tag{5.57}$$

which can be solved by a banded solver at a cost of $O(N)$ operations.

5.4 Numerical Results

Three test problems were used to test the method. The same eight quantities given in Chapter 3 are given in Tables 5.1–5.4 for u, u', v, v', v''. Note that $v = u''$.

Problem 5.1:

$$u(x) = x^2(1-x)^2\sin(\pi x).$$

Problem 5.2:

$$u(x) = [1-\cos(2\pi x)]^2.$$

Problem 5.3:

$$u(x) = e^x(x^2-x)^2.$$

For Problem 5.1, we present the results of (5.5) and (5.6) using the unperturbed Neumann boundary conditions $u_h(0) = u_h(1) = 0$ in Table 5.1, and the results using the perturbed boundary conditions given in (5.7) in Table 5.2. The results for Problems 5.2 and 5.3 are presented in Tables 5.3 and 5.4, respectively. From Table 5.1, we see that, when using the unperturbed boundary conditions, u and v exhibit suboptimal second-order global accuracy. Furthermore, neither u nor v is superconvergent at the nodal or collocation points and u' is not superconvergent at the Gauss points. We do observe superconvergence of v' at the Gauss points and v'' at the collocation points, that is, third-order and second-order accuracy, respectively. In contrast, the use of (5.7) results in optimal accuracy globally—third-order accuracy in u and v, second-order accuracy in u' and v' and first-order accuracy for v''—and the expected superconvergence results—fourth-order accuracy for the u and v approximations at the nodal and collocation points, third-order accuracy for the u' and v' approximations at the Gauss points and second-order accuracy for the v'' approximation at the collocation points. Moreover, the error is reduced for all values of N for each approximation with the exception of v' and v'' which are unchanged. The results of Problems 5.2 and 5.3 similarly indicate global optimal accuracy and the same superconvergence phenomena.

Table 5.1: Test Problem 5.1: Unperturbed BCs.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	6.0318(-3)	6.1058(-3)	6.0532(-3)	6.1072(-3)				
16	1.5280(-3)	1.5326(-3)	1.5293(-3)	1.5327(-3)	1.9809	1.9941	1.9848	1.9944
32	3.8313(-4)	3.8342(-4)	3.8321(-4)	3.8342(-4)	1.9957	1.9990	1.9967	1.9991
64	9.5851(-5)	9.5869(-5)	9.5856(-5)	9.5870(-5)	1.9990	1.9998	1.9992	1.9998
128	2.3967(-5)	2.3968(-5)	2.3967(-5)	2.3968(-5)	1.9997	1.9999	1.9998	2.0000
256	5.9920(-6)	5.9921(-6)	5.9920(-6)	5.9921(-6)	1.9999	2.0000	2.0000	2.0000
u'								
8	2.7366(-2)	2.7640(-2)	2.4719(-2)	2.7798(-2)				
16	6.5977(-3)	8.0592(-3)	6.1603(-3)	8.0404(-3)	2.0523	1.7780	2.0046	1.7896
32	1.6348(-3)	2.1576(-3)	1.5353(-3)	2.1457(-3)	2.0128	1.9012	2.0045	1.9059
64	4.0781(-4)	5.5730(-4)	3.8348(-4)	5.5341(-4)	2.0032	1.9529	2.0012	1.9550
128	1.0190(-4)	1.4157(-4)	9.5859(-5)	1.4048(-4)	2.0008	1.9770	2.0002	1.9779
256	2.5474(-5)	3.5672(-5)	2.3966(-5)	3.5388(-5)	2.0000	1.9886	1.9999	1.9891
v								
8	5.9697(-2)	5.2700(-2)	6.3095(-2)	6.2802(-2)				
16	1.2926(-2)	1.2500(-2)	1.3926(-2)	1.3898(-2)	2.2074	2.0759	2.1798	2.1760
32	3.1087(-3)	3.0822(-3)	3.2701(-3)	3.2669(-3)	2.0559	2.0198	2.0903	2.0889
64	7.6953(-4)	7.6788(-4)	7.9203(-4)	7.9163(-4)	2.0143	2.0050	2.0457	2.0450
128	1.9191(-4)	1.9180(-4)	1.9486(-4)	1.9481(-4)	2.0036	2.0013	2.0231	2.0227
256	4.7947(-5)	4.7940(-5)	4.8326(-5)	4.8320(-5)	2.0009	2.0003	2.0116	2.0114
v'								
8	1.0832	5.2543(-1)	2.2672(-1)	5.5978(-1)				
16	2.6600(-1)	1.3234(-1)	2.8659(-2)	1.3502(-1)	2.0258	1.9893	2.9839	2.0517
32	6.6204(-2)	3.3089(-2)	3.6092(-3)	3.3442(-2)	2.0064	1.9998	2.9892	2.0134
64	1.6532(-2)	8.2687(-3)	4.5311(-4)	8.3398(-3)	2.0016	2.0006	2.9938	2.0036
128	4.1319(-3)	2.0665(-3)	5.6770(-5)	2.0835(-3)	2.0004	2.0004	2.9967	2.0010
256	1.0330(-3)	5.1654(-4)	7.1047(-6)	5.2079(-4)	2.0000	2.0003	2.9983	2.0002
v''								
8		4.0669	2.9447(+1)	4.1598(+1)				
16		1.1284	1.4686(+1)	2.0792(+1)		1.8497	1.0037	1.0005
32		2.9341(-1)	7.3341	1.0391(+1)		1.9433	1.0017	1.0008
64		7.4587(-2)	3.6653	5.1938		1.9759	1.0007	1.0004
128		1.8790(-2)	1.8323	2.5965		1.9890	1.0003	1.0002
256		4.7146(-3)	9.1608(-1)	1.2982		1.9947	1.0001	1.0001

Table 5.2: Test Problem 5.1: Perturbed BCs.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	2.0406(-4)	3.1185(-4)	3.7130(-4)	3.9593(-4)				
16	1.2375(-5)	1.8731(-5)	4.1330(-5)	4.1611(-5)	4.0435	4.0573	3.1673	3.2502
32	7.6811(-7)	1.1601(-6)	4.8790(-6)	4.7481(-6)	4.0100	4.0131	3.0826	3.1316
64	4.7926(-8)	7.2287(-8)	5.9290(-7)	5.7950(-7)	4.0024	4.0044	3.0407	3.0344
128	2.9941(-9)	4.5164(-9)	7.3083(-8)	7.1663(-8)	4.0006	4.0005	3.0202	3.0155
256	1.8712(-10)	2.8224(-10)	9.0720(-9)	8.9103(-9)	4.0001	4.0002	3.0100	3.0077
u'								
8	2.4943(-2)	7.9892(-3)	5.1267(-3)	1.4970(-2)				
16	6.1615(-3)	2.2828(-3)	6.1590(-4)	3.3933(-3)	2.0173	1.8072	3.0573	2.1414
32	1.5356(-3)	6.7004(-4)	7.6211(-5)	8.0922(-4)	2.0045	1.7685	3.0146	2.0681
64	3.8360(-4)	1.7970(-4)	9.4996(-6)	1.9772(-4)	2.0011	1.8987	3.0041	2.0331
128	9.5880(-5)	4.6435(-5)	1.1858(-6)	4.8875(-5)	2.0003	1.9523	3.0021	2.0163
256	2.3969(-5)	1.1797(-5)	1.4811(-7)	1.2151(-5)	2.0001	1.9768	3.0010	2.0081
v								
8	9.8103(-3)	4.1234(-3)	1.3590(-2)	1.3283(-2)				
16	6.0291(-4)	2.7283(-4)	1.6194(-3)	1.5987(-3)	4.0243	3.9178	3.0690	3.0546
32	3.7523(-5)	1.7507(-5)	2.0003(-4)	1.9720(-4)	4.0061	3.9620	3.0172	3.0192
64	2.3427(-6)	1.1086(-6)	2.4920(-5)	2.4529(-5)	4.0015	3.9811	3.0048	3.0071
128	1.4638(-7)	6.9746(-8)	3.1103(-6)	3.0602(-6)	4.0004	3.9905	3.0022	3.0028
256	9.1481(-9)	4.3738(-9)	3.8851(-7)	3.8222(-7)	4.0001	3.9952	3.0010	3.0012
v'								
8	1.0832	5.2543(-1)	2.2672(-1)	5.5978(-1)				
16	2.6600(-1)	1.3234(-1)	2.8659(-2)	1.3502(-1)	2.0258	1.9893	2.9839	2.0517
32	6.6204(-2)	3.3089(-2)	3.6092(-3)	3.3442(-2)	2.0064	1.9998	2.9892	2.0134
64	1.6532(-2)	8.2687(-3)	4.5311(-4)	8.3398(-3)	2.0016	2.0006	2.9938	2.0036
128	4.1319(-3)	2.0665(-3)	5.6770(-5)	2.0835(-3)	2.0004	2.0004	2.9967	2.0010
256	1.0330(-3)	5.1654(-4)	7.1047(-6)	5.2079(-4)	2.0000	2.0003	2.9983	2.0002
v''								
8		4.0669	2.9447(+1)	4.1598(+1)				
16		1.1284	1.4686(+1)	2.0792(+1)		1.8497	1.0037	1.0005
32		2.9341(-1)	7.3341	1.0391(+1)		1.9433	1.0017	1.0008
64		7.4587(-2)	3.6653	5.1938		1.9759	1.0007	1.0004
128		1.8790(-2)	1.8323	2.5965		1.9890	1.0003	1.0002
256		4.7146(-3)	9.1608(-1)	1.2982		1.9947	1.0001	1.0001

Table 5.3: Test Problem 5.2.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	1.7871(-2)	2.5352(-2)	3.5373(-2)	3.6595(-2)				
16	1.0030(-3)	1.5805(-3)	3.2141(-3)	3.1395(-3)	4.1552	4.0037	3.4602	3.5430
32	6.1692(-5)	9.8637(-5)	3.6068(-4)	3.5594(-4)	4.0231	4.0021	3.1556	3.1408
64	3.8542(-6)	6.1619(-6)	4.3356(-5)	4.2671(-5)	4.0006	4.0007	3.0564	3.0603
128	2.4099(-7)	3.8507(-7)	5.2977(-6)	5.2136(-6)	3.9994	4.0002	3.0328	3.0329
256	1.5060(-8)	2.4066(-8)	6.5518(-7)	6.4465(-7)	4.0002	4.0000	3.0154	3.0157
u'								
8	2.0316	8.3257(-1)	4.8882(-1)	1.1884				
16	4.5200(-1)	2.2226(-1)	6.1077(-2)	2.3912(-1)	2.1683	1.9053	3.0006	2.3132
32	1.1018(-1)	5.5592(-2)	7.6302(-3)	5.6485(-2)	2.0364	1.9993	3.0008	2.0818
64	2.7655(-2)	1.3809(-2)	9.5357(-4)	1.3982(-2)	1.9943	2.0093	3.0003	2.0143
128	6.9035(-3)	3.4529(-3)	1.1919(-4)	3.4836(-3)	2.0021	1.9997	3.0001	2.0049
256	1.7259(-3)	8.6298(-4)	1.4898(-5)	8.7026(-4)	2.0000	2.0004	3.0000	2.0011
v								
8	3.1645	1.2124	3.2889	3.2562				
16	1.7782(-1)	7.7977(-2)	3.5410(-1)	3.4714(-1)	4.1535	3.9587	3.2154	3.2296
32	1.0830(-2)	4.8584(-3)	4.2502(-2)	4.1823(-2)	4.0373	4.0045	3.0586	3.0532
64	6.7252(-4)	3.0323(-4)	5.2497(-3)	5.1666(-3)	4.0093	4.0020	3.0172	3.0170
128	4.1965(-5)	1.8944(-5)	6.5367(-4)	6.4320(-4)	4.0023	4.0006	3.0056	3.0059
256	2.6218(-6)	1.1839(-6)	8.1600(-5)	8.0280(-5)	4.0006	4.0001	3.0019	3.0022
v'								
8	2.4159	1.0655	7.0443	1.3356				
16	5.6670	2.7974	8.3544	2.9314	2.0919	1.9294	3.0759	2.1878
32	1.3949	6.9845	1.0291	7.0877	2.0224	2.0019	3.0212	2.0482
64	3.4739	1.7411	1.2815(-1)	1.7560	2.0056	2.0042	3.0055	2.0130
128	8.6764(-1)	4.3451(-1)	1.6003(-2)	4.3785(-1)	2.0014	2.0025	3.0014	2.0038
256	2.1705(-1)	1.0853(-1)	1.9999(-3)	1.0943(-1)	1.9991	2.0014	3.0003	2.0004
v''								
8		9.8042(+2)	6.2458(+3)	8.7662(+3)				
16		3.1571(+2)	3.1051(+3)	4.3857(+3)		1.6348	1.0083	0.9991
32		8.3559(+1)	1.5447(+3)	2.1868(+3)		1.9177	1.0073	1.0040
64		2.1183(+1)	7.7063(+2)	1.0917(+3)		1.9799	1.0032	1.0022
128		5.3141	3.8494(+2)	5.4544(+2)		1.9950	1.0014	1.0011
256		1.3297	1.9248(+2)	2.7276(+2)		1.9988	0.9999	0.9998

Table 5.4: Test Problem 5.3.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	2.9443(-5)	3.6619(-4)	8.3389(-4)	8.3740(-4)				
16	1.9083(-6)	2.5053(-5)	1.0100(-4)	9.8476(-5)	3.9476	3.8695	3.0456	3.0881
32	1.1975(-7)	1.6335(-6)	1.2330(-5)	1.2043(-5)	3.9942	3.9389	3.0341	3.0316
64	7.4979(-9)	1.0421(-7)	1.5199(-6)	1.4897(-6)	3.9973	3.9704	3.0201	3.0151
128	4.6875(-10)	6.5794(-9)	1.8857(-7)	1.8516(-7)	3.9996	3.9854	3.0108	3.0082
256	2.9302(-11)	4.1328(-10)	2.3479(-8)	2.3077(-8)	3.9998	3.9928	3.0056	3.0043
u'								
8	6.2514(-2)	2.3326(-2)	5.9698(-3)	3.4245(-2)				
16	1.5853(-2)	6.8496(-3)	8.2004(-4)	8.3700(-3)	1.9795	1.7678	2.8639	2.0326
32	3.9772(-3)	1.8487(-3)	1.0713(-4)	2.0546(-3)	1.9949	1.8895	2.9364	2.0264
64	9.9517(-4)	4.7977(-4)	1.3680(-5)	5.0804(-4)	1.9987	1.9461	2.9692	2.0159
128	2.4885(-4)	1.2218(-4)	1.7281(-6)	1.2625(-4)	1.9997	1.9734	2.9848	2.0086
256	6.2215(-5)	3.0825(-5)	2.1714(-7)	3.1466(-5)	1.9999	1.9868	2.9925	2.0045
v								
8	2.1693(-3)	9.8179(-4)	9.5847(-3)	9.3264(-3)				
16	1.3681(-4)	6.3901(-5)	1.2915(-3)	1.2577(-3)	3.9870	3.9415	2.8917	2.8906
32	8.5699(-6)	4.0843(-6)	1.6723(-4)	1.6371(-4)	3.9967	3.9677	2.9491	2.9415
64	5.3592(-7)	2.5829(-7)	2.1263(-5)	2.0867(-5)	3.9992	3.9831	2.9754	2.9718
128	3.3500(-8)	1.6240(-8)	2.6802(-6)	2.6335(-6)	3.9998	3.9913	2.9879	2.9862
256	2.0938(-9)	1.0181(-9)	3.3641(-7)	3.3076(-7)	3.9999	3.9956	2.9940	2.9931
v'								
8	9.1809(-1)	3.9065(-1)	5.1446(-2)	4.8664(-1)				
16	2.2993(-1)	1.0632(-1)	6.5724(-3)	1.1896(-1)	1.9974	1.8774	2.9686	2.0324
32	5.7507(-2)	2.7669(-2)	8.2998(-4)	2.9377(-2)	1.9994	1.9421	2.9853	2.0178
64	1.4378(-2)	7.0533(-3)	1.0426(-4)	7.2970(-3)	1.9998	1.9719	2.9929	2.0093
128	3.5947(-3)	1.7804(-3)	1.3065(-5)	1.8183(-3)	2.0000	1.9861	2.9965	2.0047
256	8.9868(-4)	4.4721(-4)	1.6350(-6)	4.5381(-4)	2.0000	1.9931	2.9983	2.0024
v''								
8		9.7894(-1)	2.3864(+1)	3.4025(+1)				
16		2.6124(-1)	1.2342(+1)	1.7541(+1)		1.9059	0.9512	0.9559
32		6.7465(-2)	6.2735	8.9027		1.9532	0.9763	0.9784
64		1.7141(-2)	3.1623	4.4845		1.9766	0.9883	0.9893
128		4.3201(-3)	1.5875	2.2505		1.9883	0.9942	0.9947
256		1.0844(-3)	7.9537(-1)	1.1273		1.9942	0.9971	0.9974

Chapter 6

AUXILIARY BIHARMONIC PROBLEM

This chapter extends the method developed in Chapters 4 and 5 to the solution of

$$-\Delta^2 u(x,y) = -f(x,y) \quad \text{in} \quad \Omega,$$

$$u = 0 \quad \text{on} \quad \partial\Omega, \qquad D_y u = 0 \quad \text{on} \quad \Gamma_x, \qquad \Delta u = 0 \quad \text{on} \quad \Gamma_y, \tag{6.1}$$

where

$$\Gamma_x = \{(x,\alpha) : x \in [0,1], \alpha = 0,1\}, \quad \text{and} \quad \Gamma_y = \{(\alpha, y) : y \in [0,1], \alpha = 0,1\}.$$

6.1 Derivation

As in Chapter 5, we reformulate (6.1) as the coupled system

$$\Delta u(x,y) = v(x,y),$$

$$\Delta v(x,y) = f(x,y), \tag{6.2}$$

and then we rewrite the boundary condition

$$\Delta u = 0 \quad \text{on} \quad \Gamma_y,$$

as

$$v = 0. \tag{6.3}$$

We then seek $u_h \in S_2^D \otimes S_2^D$ and $v_h \in S_2^D \otimes S_2$ such that

$$-\Delta u_h(\tau_i, \tau_j) + \frac{h^2}{12} D_x^2 D_y^2 u_h(\tau_i, \tau_j) + v_h(\tau_i, \tau_j) = \frac{h^2}{24} f(\tau_i, \tau_j),$$

$$-\Delta v_h(\tau_i, \tau_j) + \frac{h^2}{12} D_x^2 D_y^2 v_h(\tau_i, \tau_j) = -f(\tau_i, \tau_j) + \frac{h^2}{24} \Delta f(\tau_i, \tau_j), \tag{6.4}$$

$$i, j = 1, \ldots, N.$$

Note that (6.4) is the same as (4.11) with v in place of f.

Now we perturb the boundary conditions on Γ_x as in Chapter 5. The boundary condition $u = 0$ on $\partial\Omega$ implies $D_x^2 u = 0$ on Γ_x. Hence, on Γ_x, we have

$$v = D_x^2 u + D_y^2 u = D_y^2 u,$$

which implies that

$$D_y^3 u = D_y v, \quad \text{on } \Gamma_x. \tag{6.5}$$

Then replacing w_h with u_h in (2.10), and using (6.5) with v replaced with v_h, yields the perturbed Neumann boundary conditions

$$D_y u_h + \frac{h^2}{12} D_y v_h = 0, \quad \text{on } \Gamma_x. \tag{6.6}$$

Then setting

$$u_h(x, y) = \sum_{m=1}^{N} \sum_{n=1}^{N} u_{m,n} \mathcal{B}_m^D(x) \mathcal{B}_n^D(y), \tag{6.7}$$

$$v_h(x, y) = \sum_{m=1}^{N} \sum_{n=0}^{N+1} v_{m,n} \mathcal{B}_m^D(x) \mathcal{B}_n(y), \tag{6.8}$$

and substituting into (6.4) and rearranging, we get

$$-\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=1}^{N}\mathcal{B}_n^D(\tau_j)u_{m,n}-\sum_{m=1}^{N}\mathcal{B}_m^D(\tau_i)\sum_{n=1}^{N}[\mathcal{B}_n^D(\tau_j)]''u_{m,n}$$

$$+\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=1}^{N}[\mathcal{B}_n^D(\tau_j)]''u_{m,n}+\sum_{m=1}^{N}\mathcal{B}_m^D(\tau_i)\sum_{n=0}^{N+1}\mathcal{B}_n v_{m,n}=\frac{h^2}{24}f(\tau_i,\tau_j), \tag{6.9}$$

$$-\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=0}^{N+1}\mathcal{B}_n(\tau_j)v_{m,n}-\sum_{m=1}^{N}\mathcal{B}_m^D(\tau_i)\sum_{n=0}^{N+1}\mathcal{B}_n''(\tau_j)v_{m,n}$$

$$+\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=0}^{N+1}[\mathcal{B}_n(\tau_j)]''v_{m,n}=-f(\tau_i,\tau_j)+\frac{h^2}{24}\Delta f(\tau_i,\tau_j), \tag{6.10}$$

$$\sum_{m=1}^{N}\mathcal{B}_m^D(\tau_i)\sum_{n=1}^{N}[\mathcal{B}_n^D(\alpha)]'u_{m,n}+\frac{h^2}{12}\sum_{m=1}^{D}\mathcal{B}_m^D(\tau_i)\sum_{n=0}^{N+1}\mathcal{B}_n'(\alpha)v_{m,n}=0, \tag{6.11}$$

$$i,j=1,\ldots,N;\quad \alpha=0,1.$$

Now, using (2.5), (2.7), (2.17), and (2.18), the set of equations (6.9)–(6.11) yields the linear system

$$-\left[A^D\otimes B^D+\left(B^D-\frac{h^2}{12}A^D\right)\otimes A^D\right]\mathbf{u}+\left(B^D\otimes B\right)\mathbf{v}=\mathbf{f}_1,$$

$$-\left[A^D\otimes B+\left(B^D-\frac{h^2}{12}A^D\right)\otimes A\right]\mathbf{v}=\mathbf{f}_2, \tag{6.12}$$

$$(B^D\otimes C)\mathbf{u}+\frac{h^2}{12}(B^D\otimes D)\mathbf{v}=\mathbf{0},$$

where

$$\mathbf{u} = [u_{1,1}, \ldots, u_{1,N}, \ldots, u_{N,1}, \ldots, u_{N,N}]^T,$$

$$\mathbf{v} = [v_{1,0}, \ldots, v_{1,N+1}, \ldots, v_{N,0}, \ldots, v_{N,N+1}]^T,$$

$$\mathbf{f}_1 = \left[f_{1,1}^{(1)}, \ldots, f_{1,N}^{(1)}, \ldots, f_{N,1}^{(1)}, \ldots, f_{N,N}^{(1)}\right]^T, \quad f_{i,j}^{(1)} = \frac{h^2}{24} f(\tau_i, \tau_j),$$

$$\mathbf{f}_2 = [f_{1,1}^{(2)}, \ldots, f_{1,N}^{(2)}, \ldots, f_{N,1}^{(2)}, \ldots, f_{N,N}^{(2)}]^T, \quad f_{i,j}^{(2)} = -f(\tau_i, \tau_j) + \frac{h^2}{24} \Delta f(\tau_i, \tau_j),$$

and A^D, B^D, A, B, C, and D are given in (3.12), (3.13), (5.11), (5.12), (5.13), and (5.14), respectively.

6.2 Solving the System

Using (3.14) and (3.15) in (6.12) and simplifying, we have

$$-\frac{1}{h^2}\left[T \otimes I + \frac{1}{6}(T + 6I) \otimes T\right]\mathbf{u} + \left[\left(\frac{1}{8}T + I\right) \otimes B\right]\mathbf{v} = \mathbf{f}_1,$$

$$-\left[\frac{1}{h^2}T \otimes B + \left(\frac{1}{24}T + I\right) \otimes A\right]\mathbf{v} = \mathbf{f}_2, \tag{6.13}$$

$$\left[\left(\frac{1}{8}T + I\right) \otimes C\right]\mathbf{u} + \frac{h^2}{12}\left[\left(\frac{1}{8}T + I\right) \otimes D\right]\mathbf{v} = \mathbf{0},$$

Since (6.13) is equivalent to

$$-(Z^T \otimes I)\frac{1}{h^2}\left[T \otimes I + \frac{1}{6}(T + 6I) \otimes T\right](Z \otimes I)(Z^{-1} \otimes I)\mathbf{u}$$
$$+(Z^T \otimes I)\left[\left(\frac{1}{8}T + I\right) \otimes B\right](Z \otimes I)(Z^{-1} \otimes I)\mathbf{v} = (Z^T \otimes I)\mathbf{f}_1,$$

$$-(Z^T \otimes I)\left[\frac{1}{h^2}T \otimes B + \left(\frac{1}{24}T + I\right) \otimes A\right](Z \otimes I)(Z^{-1} \otimes I)\mathbf{v} = (Z^T \otimes I)\mathbf{f}_2, \quad (6.14)$$

$$(Z^T \otimes I)\left[\left(\frac{1}{8}T + I\right) \otimes C\right](Z \otimes I)(Z^{-1} \otimes I)\mathbf{u}$$
$$+\frac{h^2}{12}(Z^T \otimes I)\left[\left(\frac{1}{8}T + I\right) \otimes D\right](Z \otimes I)(Z^{-1} \otimes I)\mathbf{v} = \mathbf{0},$$

where Z is given in (2.13)–(2.14) and I is the $N \times N$ identity matrix, we can use (2.15) in (6.14) to obtain the system

$$-\frac{1}{h^2}\left[\Lambda_T \otimes I + \frac{1}{6}(\Lambda_T + 6I) \otimes T\right]\widetilde{\mathbf{u}} + \left[\left(\frac{1}{8}\Lambda_T + I\right) \otimes B\right]\widetilde{\mathbf{v}} = \widetilde{\mathbf{f}}_1,$$

$$-\left[\frac{1}{h^2}\Lambda_T \otimes B + \left(\frac{1}{24}\Lambda_T + I\right) \otimes A\right]\widetilde{\mathbf{v}} = \widetilde{\mathbf{f}}_2, \quad (6.15)$$

$$\left[\left(\frac{1}{8}\Lambda_T + I\right) \otimes C\right]\widetilde{\mathbf{u}} + \frac{h^2}{12}\left[\left(\frac{1}{8}\Lambda_T + I\right) \otimes D\right]\widetilde{\mathbf{v}} = \mathbf{0}.$$

where

$$\widetilde{\mathbf{u}} = (Z^{-1} \otimes I)\mathbf{u}, \quad \widetilde{\mathbf{v}} = (Z^{-1} \otimes I)\mathbf{v}, \quad \widetilde{\mathbf{f}}_1 = (Z^T \otimes I)\mathbf{f}_1, \quad \widetilde{\mathbf{f}}_2 = (Z^T \otimes I)\mathbf{f}_2.$$

Since Λ_T and I are diagonal, we can decouple (6.15) into the N linear systems

$$
\begin{gathered}
-\frac{1}{h^2}\left[\lambda_i^T I + \left(\frac{1}{6}\lambda_i^T + 1\right) T\right]\widetilde{\mathbf{u}}_\mathbf{i} + \left(\frac{1}{8}\lambda_i^T + 1\right) B\widetilde{\mathbf{v}}_\mathbf{i} = \widetilde{\mathbf{f}}_\mathbf{i}^{(1)}, \\
-\left[\frac{1}{h^2}\lambda_i^T B + \left(\frac{1}{24}\lambda_i^T + 1\right) A\right]\widetilde{\mathbf{v}}_\mathbf{i} = \widetilde{\mathbf{f}}_\mathbf{i}^{(2)}, \\
\left(\frac{1}{8}\lambda_i^T + 1\right) C\widetilde{\mathbf{u}}_\mathbf{i} + \frac{h^2}{12}\left(\frac{1}{8}\lambda_i^T + 1\right) D\widetilde{\mathbf{v}}_\mathbf{i} = \mathbf{0}, \\
i = 1, \ldots, N,
\end{gathered}
\tag{6.16}
$$

where

$$
\widetilde{\mathbf{u}}_\mathbf{i} = [\widetilde{u}_{i,1}, \ldots, \widetilde{u}_{i,N}]^T, \qquad \widetilde{\mathbf{v}}_\mathbf{i} = [\widetilde{v}_{i,1}, \ldots, \widetilde{v}_{i,N}]^T, \qquad \widetilde{\mathbf{f}} = [\widetilde{f}_{i,1}, \ldots, \widetilde{f}_{i,N}]^T.
$$

Note that the last equation of (6.16) (which corresponds to the perturbed Neumann boundary conditions) has a common factor of $\left(\frac{1}{8}\lambda_i^T + 1\right)$, which we divide out to get

$$
C\widetilde{\mathbf{u}}_\mathbf{i} + \frac{h^2}{12}D\widetilde{\mathbf{v}}_\mathbf{i} = \mathbf{0}. \tag{6.17}
$$

Now, we combine (6.16) and (6.17) into the N linear systems

$$
S_i \begin{bmatrix} \widetilde{\mathbf{u}}_\mathbf{i} \\ \widetilde{\mathbf{v}}_\mathbf{i} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{f}}_\mathbf{i}^{(1)} \\ \widetilde{\mathbf{f}}_\mathbf{i}^{(2)} \\ \mathbf{0} \end{bmatrix}, \qquad i = 1, \ldots, N, \tag{6.18}
$$

where S is the $(2N+2)\times(2N+2)$ matrix given by

$$S_i = \begin{bmatrix} -\frac{1}{h^2}\left[\lambda_i^T I + \left(\frac{1}{6}\lambda_i^T + 1\right) T\right] & \left(\frac{1}{8}\lambda_i^T + 1\right) B \\ 0 & \frac{1}{h^2}\lambda_i^T B + \left(\frac{1}{24}\lambda_i^T + 1\right) A \\ C & \frac{h^2}{12} D \end{bmatrix}.$$

This leads to the following **Matrix Decomposition Algorithm:**

Step 1. Compute

$$\widetilde{\mathbf{f}_1} = (Z^T \otimes I)\,\mathbf{f}_1, \quad \widetilde{\mathbf{f}_2} = (Z^T \otimes I)\,\mathbf{f}_2.$$

Step 2. Solve the N linear systems given in (6.18).

Step 3. Compute

$$\mathbf{u} = (Z \otimes I)\widetilde{\mathbf{u}}, \quad \mathbf{v} = (Z \otimes I)\widetilde{\mathbf{v}}.$$

Since Z is composed of sines, FFTs can be used to perform steps 1 and 3 at a cost of $O(N^2 \log N)$ operations. As written in (6.18), the coefficient matrix has no special structure, but it can be rewritten in the same form as (5.55). In this form, step 2 can be performed at a cost of $O(N^2)$ operations, leading to a total cost of $O(N^2 \log N)$ operations for the algorithm.

6.3 Existence and Uniqueness

Since both matrices Z and I are nonsingular, to show that the method produces a unique solution, all that is necessary is to show that each of the N systems described by (6.16) produce a unique solution. Clearly, setting $\mathbf{f}_2 = \mathbf{0}$, the second of (6.16) is the matrix–vector form of (5.17), and (6.17) is the matrix–vector form of (5.15). All that remains is to show that the first equation of (6.16) is the matrix-vector form of (5.16).

The first equation in (6.16) is equivalent to the first equation in (6.12). Then applying (3.14) and (3.15) to the first argument of each of the tensor products gives

$$-\left[\frac{1}{h^2}T\otimes B^D+\left(\frac{1}{8}T+I-\frac{h^2}{12}\frac{1}{h^2}T\right)\otimes A^D\right]\mathbf{u}+\left(\frac{1}{8}T+I\right)\otimes B\mathbf{v}=\mathbf{f}_1. \tag{6.19}$$

Setting $\mathbf{f}_1=\mathbf{0}$ multiplying by -1 and simplifying yields

$$\left[\frac{1}{h^2}T\otimes B^D+\left(\frac{1}{24}T+I\right)\otimes A^D\right]\mathbf{u}-\left(\frac{1}{8}T+I\right)\otimes B\mathbf{v}=\mathbf{0}. \tag{6.20}$$

As in Section 6.2, we see that this system is equivalent to

$$\left[\frac{1}{h^2}\Lambda_T\otimes B^D+\left(\frac{1}{24}\Lambda_T+I\right)\otimes A^D\right]\widetilde{\mathbf{u}}-\left(\frac{1}{8}\Lambda_T+I\right)\otimes B\widetilde{\mathbf{v}}=\mathbf{0}, \tag{6.21}$$

which can be written as the N linear systems

$$\left[\frac{1}{h^2}\lambda_i^T B^D+\left(\frac{1}{24}\lambda_i^T+1\right)A^D\right]\widetilde{\mathbf{u}}-\left(\frac{1}{8}\lambda_i^T+1\right)B\widetilde{\mathbf{v}}=0, \tag{6.22}$$

for $i=1,\ldots,N$. This equation is the matrix-vector form of (5.16).

6.4 Numerical Results

We demonstrate the accuracy of (6.4) and (6.6) with the test problem from [1] Chapter 5.

Problem 6.1:

$$u(x,y)=\sin(\pi x)\sin^2(\pi y).$$

Table 6.1 contains the results for the approximation of $u(x,y)$ and Table 6.2 contains the results of the approximation of $v(x,y)$. Both tables have the same eight values discussed in Chapter 3.

As seen in the tables, the method produces third-order accuracy globally for u and v, second-order global accuracy for u_x, u_y, u_{xy}, v_x, v_y, and v_{xy}, and first-order global accuracy for u_{xx}, u_{yy}, v_{xx}, and v_{yy}. Furthermore, the method is superconvergent for the u approximation at the nodes and collocation points, for the u_x, u_y, u_{xy} approximations at the Gauss points, and for the u_{xx} and u_{yy} approximations at the collocation points. The same is true for the v and its derivative approximations at corresponding points. Note that the error for v_{xx} and v_{yy} is rather large absolutely for any value of N, however, it is relatively small.

In comparison, the global error reported in [1] Tables 5.3 and 5.4 for u and v, respectively, is consistently smaller than the global error demonstrated by (6.4) and (6.6). This is expected as the MNCSC method is fourth-order accurate globally and our method is only third-order accurate. However, the error at nodes and collocation points is comparable for u and v to the global error. The same is true for the Gauss points of u_x, u_y, u_{xy}, v_x, v_y, and v_{xy} when compared to the global error in [1].

Table 6.1: Test Problem 6.1. $u(x,y)$

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	2.1619(-3)	1.4870(-3)	3.0721(-3)	3.0620(-3)				
16	1.3443(-4)	9.2940(-5)	3.1030(-4)	3.0894(-4)	4.0074	3.9999	3.3074	3.3090
32	8.3923(-6)	5.8068(-6)	3.4644(-5)	3.4249(-5)	4.0016	4.0005	3.1630	3.1732
64	5.2438(-7)	3.6289(-7)	4.0752(-6)	4.0174(-6)	4.0004	4.0001	3.0877	3.0917
128	3.2771(-8)	2.2680(-8)	4.9365(-7)	4.8611(-7)	4.0001	4.0000	3.0453	3.0469
u_x								
8	3.4125(-2)	2.0945(-2)	9.5336(-3)	2.5710(-2)				
16	9.7086(-3)	5.0991(-3)	9.8881(-4)	5.3048(-3)	1.8135	2.0383	3.2693	2.2769
32	2.4993(-3)	1.2650(-3)	1.1116(-4)	1.2667(-3)	1.9577	2.0111	3.1530	2.0662
64	6.2933(-4)	3.1562(-4)	1.3089(-5)	3.1768(-4)	1.9897	2.0029	3.0863	1.9954
128	1.5761(-4)	7.8866(-5)	1.5867(-6)	7.9483(-4)	1.9974	2.0007	3.0442	1.9988
u_y								
8	1.6491(-1)	7.8149(-2)	2.4851(-2)	8.4853(-1)				
16	4.0575(-2)	2.0043(-2)	3.0661(-3)	2.0555(-2)	2.0230	1.9631	3.0189	2.0455
32	1.0106(-2)	5.0379(-3)	3.8187(-4)	5.1007(-3)	2.0054	1.9922	3.0052	2.0107
64	2.5241(-3)	1.2611(-3)	4.7689(-5)	1.2728(-3)	2.0013	1.9981	3.0013	2.0026
128	6.3087(-4)	3.1538(-4)	5.9598(-6)	3.1807(-4)	2.0003	1.9995	3.0003	2.0007
u_{xy}								
8	6.5365(-1)	3.0292(-1)	7.7831(-2)	3.3403(-1)				
16	1.5971(-1)	7.8444(-2)	9.6245(-3)	8.0773(-2)	2.0330	1.9492	3.0156	2.0480
32	3.9708(-2)	1.9767(-2)	1.1994(-3)	2.0034(-2)	2.0080	1.9886	3.0044	2.0114
64	9.9135(-3)	4.9513(-3)	1.4981(-4)	4.9987(-3)	2.0020	1.9972	3.0011	2.0028
128	2.4775(-3)	1.2384(-3)	1.8723(-5)	1.2491(-3)	2.0005	1.9993	3.0003	2.0007
u_{xx}								
8		5.4877(-2)	1.1088	1.5797				
16		1.5303(-2)	5.5847(-1)	7.9241(-1)		1.8424	0.9894	0.9954
32		3.9288(-3)	2.7960(-1)	3.9635(-1)		1.9616	0.9981	0.9995
64		9.8872(-4)	1.3984(-1)	1.9819(-1)		1.9905	0.9996	0.9999
128		2.4759(-4)	6.9926(-2)	9.9096(-2)		1.9976	0.9999	1.0000
u_{yy}								
8		4.6279(-1)	4.4718	6.3359				
16		1.2402(-1)	2.2376	3.1707		1.8998	0.9989	0.9988
32		3.1534(-2)	1.1188	1.5855		1.9756	0.9999	0.9998
64		7.9163(-3)	5.5942(-1)	7.9277(-1)		1.9940	1.0000	1.0000
128		1.9811(-3)	2.7971(-1)	3.9639(-1)		1.9985	1.0000	1.0000

Table 6.2: Test Problem 6.1. $v(x, y)$

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
v								
8	7.7085(-2)	2.5317(-2)	1.1322(-1)	1.1616(-1)				
16	4.7255(-3)	1.6383(-3)	1.2865(-2)	1.2822(-2)	4.0279	3.9498	3.1376	3.1794
32	2.9391(-4)	1.0473(-4)	1.5444(-3)	1.5253(-3)	4.0070	3.9674	3.0583	3.0715
64	1.8347(-5)	6.6299(-6)	1.8985(-4)	1.8700(-4)	4.0018	3.9816	3.0241	3.0279
128	1.1463(-6)	4.1718(-7)	2.3565(-5)	2.3190(-5)	4.0004	3.9902	3.0102	3.0115
v_x								
8	9.6879(-1)	5.4996(-1)	3.5359(-1)	7.9654(-1)				
16	2.8511(-1)	1.4654(-1)	4.0522(-2)	1.5911(-1)	1.7647	1.9080	3.1253	2.3237
32	7.3860(-2)	3.7177(-2)	4.8627(-3)	3.7729(-2)	1.9487	1.9788	3.0589	2.0763
64	1.8625(-2)	9.3279(-3)	5.9736(-4)	9.4201(-3)	1.9876	1.9948	3.0251	2.0019
128	4.6662(-3)	2.3341(-3)	7.4116(-5)	2.3543(-3)	1.9969	1.9987	3.0107	2.0005
v_y								
8	8.0521	3.9290	1.2395	4.1051				
16	1.9976	9.9358(-1)	1.5169(-1)	1.0097	2.0111	1.9835	3.0305	2.0234
32	4.9841(-1)	2.4889(-1)	1.8857(-2)	2.5143(-1)	2.0029	1.9971	3.0080	2.0058
64	1.2454(-1)	6.2251(-2)	2.3559(-3)	6.2795(-2)	2.0007	1.9993	3.0007	2.0014
128	3.1131(-2)	1.5564(-2)	2.9456(-4)	1.5695(-2)	2.0002	1.9998	2.9996	2.0004
v_{xy}								
8	3.1984(+1)	1.5175(+1)	3.8819	1.6223(+1)				
16	7.8669	3.8852	4.7618(-1)	3.9715	2.0235	1.9656	3.0272	2.0303
32	1.9586	9.7634(-1)	5.9228(-2)	9.8775(-1)	2.0059	1.9925	3.0072	2.0075
64	4.8916(-1)	2.4439(-1)	7.4026(-3)	2.4662(-1)	2.0015	1.9982	3.0002	2.0019
128	1.2226(-1)	6.1117(-2)	9.2545(-4)	6.1635(-2)	2.0004	1.9996	2.9998	2.0005
v_{xx}								
8		1.7534	3.2665(+1)	4.6711(+1)				
16		4.6108(-1)	1.6516(+1)	2.3457(+1)		1.9270	0.9839	0.9938
32		1.1683(-1)	8.2763	1.1735(+1)		1.9807	0.9968	0.9992
64		2.9306(-2)	4.1403	5.8681		1.9951	0.9993	0.9998
128		7.3327(-3)	2.0704	2.9341		1.9988	0.9998	1.0000
v_{yy}								
8		2.2297(+1)	2.2049(+2)	3.1248(+2)				
16		6.0834	1.1042(+2)	1.5646(+2)		1.8739	0.9978	0.9980
32		1.5537	5.5212(+1)	7.8241(+1)		1.9691	0.9999	0.9998
64		3.9050(-1)	2.7606(+1)	3.9122(+1)		1.9923	1.0000	1.0000
128		9.7754(-2)	1.3803(+1)	1.9561(+1)		1.9981	1.0000	1.0000

Chapter 7

BIHARMONIC DIRICHLET PROBLEM

In this chapter, we seek to extend the method developed in Chapter 6 to solve (1.2), that is, the biharmonic Dirichlet problem.

7.1 Derivation

Again, we start by rewriting (1.2) as the coupled system

$$\begin{aligned} -\Delta u(x,y) &= -v(x,y), \\ -\Delta v(x,y) &= -f(x,y), \end{aligned} \tag{7.1}$$

and imposing the same boundary conditions as in (1.2). Now applying the scheme (4.11) to (7.1) and perturbing the boundary conditions as in Chapters 5 and 6, we get

$$-\Delta u_h(\tau_i,\tau_j) + \frac{h^2}{12}D_x^2D_y^2u_h(\tau_i,\tau_j) + v_h(\tau_i,\tau_j) = \frac{h^2}{24}f(\tau_i,\tau_j), \tag{7.2}$$

$$-\Delta v_h(\tau_i,\tau_j) + \frac{h^2}{12}D_x^2D_y^2v_h(\tau_i,\tau_j) = -f(\tau_i,\tau_j) + \frac{h^2}{24}\Delta f(\tau_i,\tau_j), \tag{7.3}$$

$$D_xu_h(\alpha,\tau_j) + \frac{h^2}{12}D_xv_h(\alpha,\tau_j) = 0, \tag{7.4}$$

$$D_yu_h(\tau_i,\beta) + \frac{h^2}{12}D_yv_h(\tau_i,\beta) = 0, \tag{7.5}$$

$$i,j = 1,\ldots,N; \quad \alpha,\beta = 0,1.$$

This gives us $2N^2+4N$ equations and $2N^2+4N+4$ unknowns. Therefore, we impose the additional condition

$$v_h(\alpha,\beta) = 0, \quad \alpha,\beta = 0,1, \tag{7.6}$$

giving us $2N^2+4N+4$ equations. We prove (7.6) starting from the fact that

$$v = D_x^2 u + D_y^2 u.$$

Since $u=0$ on $\partial\Omega$, we have $D_x^2 u = 0$ on Γ_x and $D_y^2 u = 0$ on Γ_y. Therefore, at the corners, we have

$$v = 0.$$

We introduce an alternative basis for S_2, $\{\widetilde{\mathcal{B}}_m\}_{m=0}^{N+1}$ where

$$\widetilde{\mathcal{B}}_m = \begin{cases} \mathcal{B}_m, & m = 0, \\ \mathcal{B}_m^D, & m = 1,2,\ldots,N, \\ \mathcal{B}_m, & m = N+1. \end{cases} \tag{7.7}$$

Note that we can convert from the new basis functions $\widetilde{\mathcal{B}}_m$ to the basis functions $\mathcal{B}_m$ using

$$[\mathcal{B}_0,\mathcal{B}_1,\ldots,\mathcal{B}_N,\mathcal{B}_{N+1}] = [\widetilde{\mathcal{B}}_0,\widetilde{\mathcal{B}}_1,\ldots,\widetilde{\mathcal{B}}_N,\widetilde{\mathcal{B}}_{N+1}]M, \tag{7.8}$$

where

$$M = \begin{bmatrix} 1 & 1 & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & 1 & 1 \end{bmatrix}_{(N+2)\times(N+2)} \tag{7.9}$$

This can be seen using the definition of $\widetilde{\mathcal{B}}_m$ and matrix-vector multiplication. It should also be noted that the matrix M is nonsingular.

Now we take

$$u_h = \sum_{m=1}^{N}\sum_{n=1}^{N} u_{m,n}\mathcal{B}_m^D(x)\mathcal{B}_n^D(y), \tag{7.10}$$

$$v_h = \sum_{m=0}^{N+1}\sum_{n=0}^{N+1} v_{m,n}\widetilde{\mathcal{B}}_m(x)\widetilde{\mathcal{B}}_n(y). \tag{7.11}$$

We rewrite v_h in the equivalent form

$$v_h = \sum_{m=1}^{N}\sum_{n=0}^{N+1} v_{m,n}\widetilde{\mathcal{B}}_m(x)\widetilde{\mathcal{B}}_n(y) + \sum_{m=0,N+1}\sum_{n=0}^{N+1} v_{m,n}\widetilde{\mathcal{B}}_m(x)\widetilde{\mathcal{B}}_n(y). \tag{7.12}$$

Then substituting (7.10) and (7.12) into (7.2), (7.3), (7.4), (7.5), and (7.6), we have

$$\begin{aligned} &-\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=1}^{N}\mathcal{B}_n^D(\tau_j)u_{m,n} - \sum_{m=1}^{N}\mathcal{B}_m^D(\tau_i)\sum_{n=1}^{N}[\mathcal{B}_n^D(\tau_j)]''u_{m,n} \\ &+\frac{h^2}{12}\sum_{m=1}^{N}[\mathcal{B}_m^D(\tau_i)]''\sum_{n=1}^{N}[\mathcal{B}_n^D(\tau_j)]''u_{m,n} + \sum_{m=1}^{N}\widetilde{\mathcal{B}}_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\tau_j)v_{m,n} \\ &+ \sum_{m=0,N+1}\widetilde{\mathcal{B}}_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\tau_j)v_{m,n} = \frac{h^2}{24}f(\tau_i,\tau_j), \end{aligned} \tag{7.13}$$

$$
\begin{aligned}
&-\sum_{m=1}^{N}\widetilde{\mathcal{B}}''_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\tau_j)v_{m,n}-\sum_{m=1}^{N}\widetilde{\mathcal{B}}_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}''_n(\tau_j)v_{m,n}\\
&+\frac{h^2}{12}\sum_{m=1}^{N}\widetilde{\mathcal{B}}''_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}''_n(\tau_j)v_{m,n}-\sum_{m=1,N+1}\widetilde{\mathcal{B}}''_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\tau_j)v_{m,n}\\
&-\sum_{m=1,N+1}\widetilde{\mathcal{B}}_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}''_n(\tau_j)v_{m,n}+\frac{h^2}{12}\sum_{m=1,N+1}\mathcal{B}''_m(\tau_i)\sum_{n=0}^{N+1}\mathcal{B}''_n(\tau_j)v_{m,n}\\
&=-f(\tau_i,\tau_j)+\frac{h^2}{24}\Delta f(\tau_i,\tau_j),
\end{aligned}
\tag{7.14}
$$

$$
\begin{aligned}
&\sum_{m=1}^{N}[\mathcal{B}^D_m(\tau_i)]'\sum_{n=1}^{N}\mathcal{B}^D_n(\tau_j)u_{m,n}+\frac{h^2}{12}\sum_{m=1}^{N}\widetilde{\mathcal{B}}'_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\tau_j)v_{m,n}\\
&+\sum_{m=1,N+1}\widetilde{\mathcal{B}}'_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\tau_j)v_{m,n}=0,
\end{aligned}
\tag{7.15}
$$

$$
\begin{aligned}
&\sum_{m=1}^{N}\mathcal{B}^D_m(\tau_i)\sum_{n=1}^{N}[\mathcal{B}^D_n(\tau_j)]'u_{m,n}+\frac{h^2}{12}\sum_{m=1}^{N}\widetilde{\mathcal{B}}_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}'_n(\tau_j)v_{m,n}\\
&+\sum_{m=1,N+1}\widetilde{\mathcal{B}}_m(\tau_i)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}'_n(\tau_j)v_{m,n}=0,
\end{aligned}
\tag{7.16}
$$

$$
\sum_{m=1}^{N}\widetilde{\mathcal{B}}_m(\alpha)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\beta)v_{m,n}+\sum_{m=1,N+1}\widetilde{\mathcal{B}}_m(\alpha)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\beta)v_{m,n}=0,
\tag{7.17}
$$

$$
i,j=1,\dots,N;\quad \alpha,\beta=0,1.
$$

Now, by definition $\widetilde{\mathcal{B}}_m=\mathcal{B}^D_m$ for $m=1,\dots,N$. So by (2.3) and (2.6), equation (7.17) reduces to

$$
\sum_{m=1,N+1}\widetilde{\mathcal{B}}_m(\alpha)\sum_{n=0}^{N+1}\widetilde{\mathcal{B}}_n(\beta)v_{m,n}=0,\qquad \alpha,\beta=0,1.
\tag{7.18}
$$

Upon using (2.5), (2.7), (2.17) and (2.18), (7.13)–(7.16), and (7.18) become the linear system of equations

$$-\left[A^D \otimes B^D + \left(B^D - \frac{h^2}{12}A^D\right) \otimes A^D\right] \mathbf{u} + \left(B^D \otimes \widetilde{B}\right) \mathbf{v} + \left(E \otimes \widetilde{B}\right) \mathbf{v}^* = \mathbf{f}_1, \quad (7.19)$$

$$-\left[A^D \otimes \widetilde{B} + \left(B^D - \frac{h^2}{12}A^D\right) \otimes \widetilde{A}\right] \mathbf{v} - \left[F \otimes \widetilde{B} + \left(E - \frac{h^2}{12}F\right) \otimes \widetilde{A}\right] \mathbf{v}^* = \mathbf{f}_2, \quad (7.20)$$

$$\left(B^D \otimes C\right) \mathbf{u} + \frac{h^2}{12}\left(B^D \otimes \widetilde{D}\right) \mathbf{v} + \frac{h^2}{12}\left(E \otimes \widetilde{D}\right) \mathbf{v}^* = \mathbf{0}, \quad (7.21)$$

$$\left(C \otimes B^D\right) \mathbf{u} + \frac{h^2}{12}\left(C \otimes \widetilde{B}\right) \mathbf{v} + \frac{h^2}{12}\left(G \otimes \widetilde{B}\right) \mathbf{v}^* = \mathbf{0}, \quad (7.22)$$

$$\left(H \otimes J\right) \mathbf{v}^* = \mathbf{0}, \quad (7.23)$$

where

$$\mathbf{u} = \left[u_{1,1}, \ldots, u_{1,N}, \ldots, u_{N,1}, \ldots, u_{N,N}\right]^T,$$

$$\mathbf{v} = \left[v_{1,0}, \ldots, v_{1,N+1}, \ldots, v_{N,0}, \ldots, v_{N,N+1}\right]^T,$$

$$\mathbf{v}^* = \left[v_{0,0}, \ldots, v_{0,N+1}, v_{N+1,0}, \ldots, v_{N+1,N+1}\right]^T,$$

$$\mathbf{f}_1 = \left[f^{(1)}_{1,1}, \ldots, f^{(1)}_{1,N}, \ldots, f^{(1)}_{N,1}, \ldots, f^{(1)}_{N,N}\right]^T, \quad f^{(1)}_{i,j} = \frac{h^2}{24} f(\tau_i, \tau_j),$$

$$\mathbf{f}_2 = [f^{(2)}_{1,1}, \dots, f^{(2)}_{1,N}, \dots, f^{(2)}_{N,1}, \dots, f^{(2)}_{N,N}]^T, \quad f^{(2)}_{i,j} = -f(\tau_i, \tau_j) + \frac{h^2}{24}\Delta f(\tau_i, \tau_j),$$

$$\widetilde{A} = \frac{1}{h^2}\begin{bmatrix} 1 & -3 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -3 & 1 \end{bmatrix}_{N\times(N+2)},$$

$$\widetilde{B} = \frac{1}{8}\begin{bmatrix} 1 & 5 & 1 & & & & \\ & 1 & 6 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 6 & 1 & \\ & & & & 1 & 5 & 1 \end{bmatrix}_{N\times(N+2)}$$

$$\widetilde{D} = \frac{1}{h}\begin{bmatrix} -1 & 2 & 0 & \dots & 0 \\ 0 & \dots & 0 & -2 & 1 \end{bmatrix}_{2\times(N+2)},$$

$$E = \frac{1}{8}\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix}_{N\times 2} \qquad F = \frac{1}{h^2}\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix}_{N\times 2}$$

$$G = \frac{1}{h}\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}_{2\times 2}, \qquad H = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2\times 2},$$

$$J = \frac{1}{2}\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}_{2\times(N+2)},$$

and A^D, B^D, and C are given in (3.12), (3.13), and (5.13), respectively.

7.2 Solving the System

We seek to solve (7.19)–(7.23) using the Schur complement approach as in [1]. We begin by rewriting (7.19)–(7.23) as

$$S_{11}\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + S_{12}\mathbf{v}^* = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{0} \end{bmatrix} \equiv \mathbf{g}, \tag{7.24}$$

$$S_{21}\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + S_{22}\mathbf{v}^* = \mathbf{0}, \tag{7.25}$$

where S_{11} is the $(2N^2+2N)\times(2N^2+2N)$ matrix

$$S_{11} = \begin{bmatrix} -A^D \otimes B^D - \left(B^D - \frac{h^2}{12}A^D\right) \otimes A^D & B^D \otimes \widetilde{B} \\ 0 & -A^D \otimes \widetilde{B} - \left(B^D - \frac{h^2}{12}A^D\right) \otimes \widetilde{A} \\ B^D \otimes C & \frac{h^2}{12}B^D \otimes \widetilde{D} \end{bmatrix},$$

S_{12} is the $(2N^2+2N)\times(2N+4)$ matrix

$$S_{12}=\begin{bmatrix} E\otimes\widetilde{B} \\ \\ -F\otimes\widetilde{B}+\left(E-\frac{h^2}{12}F\right)\otimes\widetilde{A}\end{bmatrix},$$

and S_{21}, S_{22} are, respectively, the $(N^2+2N)\times(2N^2+2N)$ and $(2N+4)\times(2N+4)$ matrices

$$S_{21}=\begin{bmatrix} C\otimes B^D & \frac{h^2}{12}C\otimes\widetilde{B} \\ \\ 0 & 0\end{bmatrix},\qquad S_{22}=\begin{bmatrix}\frac{h^2}{12}G\otimes\widetilde{B} \\ \\ H\otimes J\end{bmatrix}.$$

Now, using (3.14) and (3.15), the matrix S_{11} can be rewritten as

$$S_{11}=\begin{bmatrix} -\frac{1}{h^2}\left[T\otimes I+\frac{1}{6}(T+6I)\otimes T\right] & \left(\frac{1}{8}T+I\right)\otimes\widetilde{B} \\ \\ 0 & -\left[\frac{1}{h^2}T\otimes\widetilde{B}+\left(\frac{1}{24}T+I\right)\otimes\widetilde{A}\right] \\ \\ \left(\frac{1}{8}T+I\right)\otimes C & \frac{h^2}{12}\left(\frac{1}{8}T+I\right)\otimes\widetilde{D}\end{bmatrix}.$$

This is the same coefficient matrix as in the system (6.13) with the matrices A, B, and D replaced by $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{D}$, respectively. Using (7.8), we have

$$S_{11}^{(6)}=S_{11}R, \tag{7.26}$$

where

$$R = \begin{bmatrix} I \otimes I & \mathbf{0} \\ \mathbf{0} & I \otimes M \end{bmatrix}, \tag{7.27}$$

$S_{11}^{(6)}$ is the coefficient matrix in (6.13), M is given in (7.9), and I is the $N \times N$ identity matrix. Note that since M is nonsingular, the matrix R is nonsingular, and we know from Chapter 6 that the matrix $S_{11}^{(6)}$ is nonsingular. Therefore, the matrix S_{11} is nonsingular. From (7.24),

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = S_{11}^{-1} \left(\mathbf{g} - S_{12} \mathbf{v}^* \right). \tag{7.28}$$

Then substituting into (7.25), we have

$$S_{21} S_{11}^{-1} \left(\mathbf{g} - S_{12} \mathbf{v}^* \right) + S_{22} \mathbf{v}^* = \mathbf{0}, \tag{7.29}$$

from which it follows that

$$\left(S_{21} S_{11}^{-1} S_{12} - S_{22} \right) \mathbf{v}^* = S_{21} S_{11}^{-1} \mathbf{g}. \tag{7.30}$$

This leads to the following algorithm:

Step 1. Compute

$$\mathbf{r} = S_{21} S_{11}^{-1} \mathbf{g}.$$

Step 2. Solve

$$\left(S_{21} S_{11}^{-1} S_{12} - S_{22} \right) \mathbf{v}^* = \mathbf{r}.$$

Step 3. Compute $\mathbf{u}$ and $\mathbf{v}$ using

$$S_{11} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{g} - S_{12} \mathbf{v}^*.$$

The matrix $S_{11}^{-1}S_{12}$ is formed one column at a time and the vector $S_{11}^{-1}\mathbf{g}$ is computed similarly using the **Matrix Decomposition Algorithm** from Chapter 6. If implemented with FFTs, the MDA can be performed at a cost of $O(N^2 \log N)$ operations. This means that step 2 (the dominant cost) can be performed at a cost of $O(N^3 \log N)$ operations.

7.3 Numerical Results

Five test problems are considered to demonstrate the accuracy of (7.2)–(7.6). In the first three problems, $u(x,y)$ is symmetric, that is

$$u(x,y) = u(y,x),$$

thus, the results for the y-derivatives are exactly the same as the corresponding x-derivatives, so only the x-derivatives are given in the tables. For the last two problems, the closed form solution is not known.

Problem 7.1:

$$u(x,y) = [1 - \cos(2\pi x)][1 - \cos(2\pi y)].$$

Problem 7.2:

$$u(x,y) = \sin^2(2\pi x)\sin^2(2\pi y).$$

Problem 7.3:

$$u(x,y) = e^{xy}(x^2 - x)^2(y^2 - y)^2.$$

Problem 7.1 was used in [1] and [7] for the biharmonic Dirichlet problem. The results for u and v are given in Table 7.1 and Table 7.2, respectively. The results of Problem 7.2 for u and v are presented in Tables 7.3 and 7.4, respectively. For Problem 7.1 and 7.2, the results are globally optimal for u and v, that is, they exhibit third-order accuracy for u and v, second-order accuracy for u_x, u_{xy}, v_x, and v_{xy}, and first-order accuracy for u_{xx} and v_{xx}. Furthermore, we see that for u and v the approximations at the nodal and collocation points are fourth-order accurate, the approximations of u_x, u_{xy}, and v_x are third-order accurate at the Gauss points, and the u_{xx} and v_{xx} approximations are second-order accurate at the collocation points. Based on the results of previous chapters, we expect third-order accuracy at the Gauss points for v_{xy}, but this was not observed.

For Problem 7.1 in comparison, the maximum global error reported in [1] is smaller than the global error of our method since the MNCSC method is globally fourth-order accurate. However, the maximum error at the nodal and collocation points of u and v and the Gauss points of u_x, u_{xy}, and v_x are comparable to the global error in [1].

Problem 7.3 was used in [11]. The results are given in Table 7.5 for u and Table 7.6 for v. Here, the accuracy of (7.2)–(7.6) is different. The results exhibit the same globally optimal behavior for u as in Problems 7.1 and 7.2. Also, u, u_x, u_{xy}, and u_{xx} demonstrate the same superconvergence results as in Problems 7.1 and 7.2. However, v and its derivatives have none of the superconvergence results that would be expected. Furthermore, v_{xy} is globally first-order accurate, when second-order accuracy is expected. A similar loss of accuracy was observed in [2] when the MNCSC method was generalized to non-homgeneous boundary conditions and Abushama and Bialecki state that this phenomenon has also been observed in the method of [7].

Problem 7.4:

$$f(x, y) = 1, \quad x, y \in \Omega.$$

This problem corresponds to the bending of a square clamped plate under a uniform load [7]. Table 7.7 gives the computed values of deflection $u(0.5, 0.5)$ and the bending moments $M_x(1, 0.5)$ and $M_y(0.5, 1)$, where $M_x = -\Delta u$ and $M_y = -\Delta u$ on the vertical and horizontal sides of $\partial\Omega$, respectively. Comparatively, [7, Table 4] has the values $u(0.5, 0.5) = 0.0126531908$, $M_x(1, 0.5) = -0.0513337647$, and $M_y(0.5, 1) = -0.0513337647$ for $N = 128$. We observe comparable results.

Problem 7.5:

$$f(x,y) = \begin{cases} 1/(4h^2), & \text{if } |x-1/2| \le h \text{ and } |y-1/2| \le h, \\ 0, & \text{otherwise.} \end{cases}$$

This problem models the bending of a square clamped plate under a load of 1 concentrated at the center [7]. The computed values of the deflection $u(0.5, 0.5)$ and the bending moment $M_y(0.5, 1)$ are presented in Table 7.8. In [7], the corresponding values are reported in Table 6 as $u(0.5, 0.5) = 0.00560424025$ and $M_y(0.5, 1) = -0.125750723$ for $N = 128$. We again see comparable results.

Table 7.1: Test Problem 7.1 $u(x,y)$.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	1.5762(-2)	5.7382(-3)	1.7880(-2)	1.8282(-2)				
16	9.8706(-4)	3.6825(-4)	1.7189(-3)	1.7329(-3)	3.9971	3.9619	3.3788	3.3992
32	6.1707(-5)	2.3169(-5)	1.8454(-4)	1.8371(-4)	3.9996	3.9904	3.2194	3.2377
64	3.8569(-6)	1.4506(-6)	2.1359(-5)	2.1147(-5)	3.9999	3.9975	3.1111	3.1189
128	2.4106(-7)	9.0705(-8)	2.5651(-6)	2.5314(-6)	4.0000	3.9993	3.0578	3.0624
u_x								
8	6.3436(-1)	3.1352(-1)	1.0254(-1)	3.3879(-1)				
16	1.6078(-1)	8.0273(-2)	1.2370(-2)	8.1114(-2)	1.9802	1.9656	3.0514	2.0624
32	4.0329(-2)	2.0158(-2)	1.5308(-3)	2.0333(-2)	1.9952	1.9936	3.0144	1.9961
64	1.0090(-2)	5.0448(-3)	1.9086(-4)	5.0870(-3)	1.9988	1.9985	3.0037	1.9989
128	2.5231(-3)	1.2615(-3)	2.3842(-5)	1.2720(-3)	1.9997	1.9996	3.0009	1.9997
u_{xy}								
8	4.2526	1.8436	3.2164(-1)	2.1601				
16	1.0264	4.9596(-1)	3.9644(-2)	5.1818(-1)	2.0507	1.8942	3.0203	2.0596
32	2.5440(-1)	1.2612(-1)	4.8802(-3)	1.2829(-1)	2.0125	1.9754	3.0221	2.0140
64	6.3463(-2)	3.1664(-2)	6.0467(-4)	3.1996(-2)	2.0031	1.9939	3.0127	2.0034
128	1.5857(-2)	7.9244(-3)	7.5238(-5)	7.9943(-3)	2.0008	1.9985	3.0066	2.0009
u_{xx}								
8		1.7848	1.7788(+1)	2.5282(+1)				
16		4.9434(-1)	8.9421	1.2680(+1)		1.8522	0.9922	0.9956
32		1.2614(-1)	4.4744	6.3418		1.9705	0.9989	0.9996
64		3.1669(-2)	2.2376	3.1710		1.9938	0.9998	0.9999
128		7.9249(-3)	1.1188	1.5855		1.9986	0.9999	1.0000

Table 7.2: Test Problem 7.1 $v(x,y)$.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
v								
8	7.3662(-1)	2.9702(-1)	7.7105(-1)	8.0394(-1)				
16	4.5314(-2)	2.1486(-2)	8.1849(-2)	8.2819(-2)	4.0229	3.7891	3.2358	3.2791
32	2.8198(-3)	1.3675(-3)	9.2347(-3)	9.2252(-3)	4.0063	3.9738	3.1478	3.1663
64	1.7604(-4)	8.5423(-5)	1.1031(-3)	1.0931(-3)	4.0016	4.0008	3.0655	3.0771
128	1.0999(-5)	5.4275(-6)	1.3474(-4)	1.3303(-4)	4.0004	3.9763	3.0333	3.0386
v_x								
8	3.7045(+1)	1.8425(+1)	6.1307	2.0498(+1)				
16	9.4960	4.7413	7.6676(-1)	4.7880	1.9639	1.9583	2.9992	2.0980
32	2.3867	1.1928	9.6516(-2)	1.2031	1.9923	1.9909	2.9899	1.9927
64	5.9744(-1)	2.9868(-1)	1.1791(-2)	3.0118(-1)	1.9982	1.9977	3.0331	1.9981
128	1.4941(-1)	7.4701(-2)	1.4463(-3)	7.5321(-2)	1.9995	1.9994	3.0273	1.9995
v_{xy}								
8	3.3374(+2)	1.4710(+2)	6.0225(+1)	1.6898(+2)				
16	8.0965(+1)	3.9241(+1)	9.7382	4.0843(+1)	2.0433	1.9063	2.6286	2.0487
32	2.0082(+1)	9.9626	1.8723	1.0125(+1)	2.0114	1.9778	2.3789	2.0121
64	5.0106	2.5003	4.3500(-1)	2.5261	2.0029	1.9944	2.1057	2.0030
128	1.2520	6.2570(-1)	1.0819(-1)	6.3119(-1)	2.0007	1.9986	2.0075	2.0007
v_{xx}								
8		1.0193(+2)	1.0502(+3)	1.4956(+3)				
16		2.8842(+1)	5.2925(+2)	7.5082(+2)		1.8213	0.9887	0.9942
32		7.4348	2.6493(+2)	3.7554(+2)		1.9558	0.9984	0.9995
64		1.8729	1.3250(+2)	1.8778(+2)		1.9890	0.9996	0.9999
128		4.6913(-1)	6.6254(+1)	9.3892(+1)		1.9972	0.9999	1.0000

Table 7.3: Test Problem 7.2 $u(x, y)$.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	5.0092(-2)	1.9631(-2)	4.0108(-2)	4.8745(-2)				
16	3.1276(-3)	1.4144(-3)	4.1601(-3)	4.2504(-3)	4.0015	3.7949	3.2692	3.5196
32	1.9548(-4)	9.1689(-5)	4.0778(-4)	4.1095(-4)	4.0000	3.9473	3.3507	3.3706
64	1.2320(-5)	5.7859(-6)	4.5033(-5)	4.4816(-5)	3.9880	3.9861	3.1787	3.1969
128	7.7153(-7)	3.6255(-7)	5.2686(-6)	5.2153(-6)	3.9971	3.9963	3.0955	3.1032
u_x								
8	1.2559	5.7200(-1)	4.9797(-1)	7.6096(-1)				
16	3.2141(-1)	1.6046(-1)	5.4574(-2)	1.7265(-1)	1.9662	1.8338	3.1898	2.1400
32	8.0654(-2)	4.0368(-2)	6.3933(-3)	4.0839(-2)	1.9946	1.9909	3.0936	2.0798
64	2.0181(-2)	1.0093(-2)	7.7851(-4)	1.0183(-2)	1.9988	1.9998	3.0378	2.0037
128	5.0462(-3)	2.5233(-3)	9.6252(-5)	2.5445(-3)	1.9997	2.0000	3.0158	2.0007
u_{xy}								
8	2.0071(+1)	5.7819	3.0367	1.0714(+1)	2.0138			
16	4.2726	1.9355	3.9774(-1)	2.1789	2.2319	1.5789	2.9326	2.2978
32	1.0277	5.0266(-1)	4.4041(-2)	5.1938(-1)	2.0558	1.9450	3.1749	2.0688
64	2.5447(-1)	1.2656(-1)	5.1418(-3)	1.2836(-1)	2.0138	1.9898	3.0985	2.0165
128	6.3468(-2)	3.1692(-2)	6.2060(-4)	3.2001(-2)	2.0034	1.9976	3.0506	2.0041
u_{xx}								
8		5.2149	3.3703(+1)	4.8557(+1)				
16		1.8390	1.7791(+1)	2.5285(+1)		1.5038	0.9217	0.9414
32		4.9789(-1)	8.9420	1.2680(+1)		1.8850	0.9925	0.9957
64		1.2636(-1)	4.4744	6.3418		1.9783	0.9989	0.9996
128		3.1683(-2)	2.2376	3.1710		1.9958	0.9998	0.9999

Table 7.4: Test Problem 7.2 $v(x, y)$

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
v								
8	1.1952(+1)	2.2826	8.9790	1.1408(+1)				
16	7.1282(-1)	2.9491(-1)	7.7793(-1)	8.0384(-1)	4.0676	2.9523	3.5288	3.8270
32	4.3802(-2)	2.1478(-2)	8.2163(-2)	8.3116(-2)	4.0245	3.7794	3.2431	3.2737
64	2.7251(-3)	1.3680(-3)	9.2504(-3)	9.2403(-3)	4.0066	3.9727	3.1509	3.1691
128	1.7012(-4)	8.5886(-5)	1.1045(-3)	1.0946(-3)	4.0017	3.9935	3.0661	3.0776
v_x								
8	2.5508(+2)	1.2632(+2)	1.1491(+2)	1.8753(+2)				
16	7.4434(+1)	3.7399(+1)	1.2514(+1)	4.1513(+1)	1.7769	1.7560	3.1990	2.1755
32	1.9014(+1)	9.5249	1.5620	9.6163	1.9689	1.9732	3.0021	2.1100
64	4.7748	2.3886	1.9459(-1)	2.4076	1.9935	1.9955	3.0048	1.9979
128	1.1950	5.9756(-1)	2.3673(-2)	6.0245(-1)	1.9985	1.9990	3.0392	1.9987
v_{xy}								
8	5.9084(+3)	1.7512(+3)	1.1787(+3)	3.1230(+3)				
16	1.3373(+3)	6.0310(+2)	2.4532(+2)	6.7822(+2)	2.1435	1.5379	2.2644	2.2031
32	3.2401(+2)	1.5815(+2)	3.9318(+1)	1.6352(+2)	2.0452	1.9311	2.6414	2.0523
64	8.0338(+1)	3.9933(+1)	7.5032	4.0510(+1)	2.0119	1.9857	2.3896	2.0131
128	2.0043(+1)	1.0007(+1)	1.7395	1.0105(+1)	2.0030	1.9966	2.1089	2.0033
v_{xx}								
8		1.0246(+3)	7.6530(+3)	1.1209(+4)				
16		4.0989(+2)	4.2021(+3)	5.9836(+3)		1.3218	0.8649	0.9056
32		1.1544(+2)	2.1171(+3)	3.0033(+3)		1.8281	0.9890	0.9945
64		2.9741(+1)	1.0597(+3)	1.5022(+3)		1.9566	0.9984	0.9995
128		7.4918	5.3000(+2)	7.5112(+2)		1.9891	0.9996	0.9999

Table 7.5: Test Problem 7.3 $u(x, y)$.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
u								
8	2.1741(-6)	1.1925(-5)	2.8637(-5)	2.8722(-5)				
16	1.6898(-7)	7.5968(-7)	3.3899(-6)	3.2985(-6)	3.6854	3.9725	3.0786	3.1223
32	8.6746(-9)	4.8003(-8)	4.1226(-7)	4.0316(-7)	4.2839	3.9842	3.0396	3.0324
64	5.2297(-10)	3.0080(-9)	5.0841(-8)	4.9871(-8)	4.0520	3.9963	3.0195	3.0151
128	3.2677(-11)	1.8811(-10)	6.3143(-9)	6.2026(-9)	4.0004	3.9992	3.0093	3.0073
u_x								
8	2.1239(-3)	8.1773(-4)	1.9743(-4)	1.1796(-3)				
16	5.3353(-4)	2.3524(-4)	2.4769(-5)	2.8162(-4)	1.9931	1.7975	2.9947	2.0665
32	1.3354(-4)	6.2885(-5)	3.0438(-6)	6.8771(-5)	1.9983	1.9034	3.0246	2.0339
64	3.3380(-5)	1.6211(-5)	3.7520(-7)	1.7006(-5)	2.0002	1.9557	3.0201	2.0158
128	8.3463(-6)	4.1134(-6)	4.6503(-8)	4.2291(-6)	1.9998	1.9786	3.0123	2.0076
u_{xy}								
8	9.5024(-3)	4.2115(-3)	1.4110(-3)	5.6978(-3)				
16	2.3022(-3)	1.1783(-3)	1.6803(-4)	1.2405(-3)	2.0453	1.8377	3.0699	2.1995
32	5.8230(-4)	2.8993(-4)	2.4146(-5)	3.0479(-4)	1.9832	2.0229	2.7988	2.0250
64	1.4553(-4)	7.2695(-5)	3.5543(-6)	7.4864(-5)	2.0004	1.9958	2.7642	2.0255
128	3.6378(-5)	1.8185(-5)	4.7944(-7)	1.8532(-5)	2.0002	1.9991	2.8901	2.0142
u_{xx}								
8		3.6390(-3)	5.3733(-2)	7.6905(-2)				
16		9.4099(-4)	2.8205(-2)	4.0144(-2)		1.9513	0.9298	0.9379
32		2.3905(-4)	1.4450(-2)	2.0521(-2)		1.9769	0.9649	0.9681
64		6.0405(-5)	7.3126(-3)	1.0374(-2)		1.9846	0.9826	0.9841
128		1.5201(-5)	3.6786(-3)	5.2157(-3)		1.9905	0.9912	0.9920

Table 7.6: Test Problem 7.3 $v(x,y)$.

N	Maximum Absolute Error				Rate of Convergence			
	$E_n(N)$	$E_c(N)$	$E_G(N)$	$E_g(N)$	$R_n(N)$	$R_c(N)$	$R_G(N)$	$R_g(N)$
v								
8	1.6944(-3)	2.9368(-3)	4.9616(-3)	4.9505(-3)				
16	2.9854(-4)	4.3356(-4)	7.2104(-4)	7.3061(-4)	2.5048	2.7600	2.7826	2.7604
32	4.1354(-5)	5.3536(-5)	9.3431(-5)	9.4936(-5)	2.8518	3.0176	2.9481	2.9441
64	5.1955(-6)	6.3976(-6)	1.1759(-5)	1.1942(-5)	2.9927	3.0649	2.9902	2.9910
128	6.4294(-7)	7.7134(-7)	1.4706(-6)	1.4922(-6)	3.0145	3.0521	2.9992	3.0005
v_x								
8	2.0471(-1)	4.3682(-2)	4.3103(-2)	9.9880(-2)				
16	5.5496(-2)	1.4201(-2)	8.8333(-3)	2.9247(-2)	1.8831	1.6210	2.2868	1.7719
32	1.4015(-2)	4.0907(-3)	1.9112(-3)	7.6124(-3)	1.9854	1.7956	2.2085	1.9419
64	3.4934(-3)	1.1077(-3)	4.6438(-4)	1.9211(-3)	2.0043	1.8848	2.0411	1.9864
128	8.7043(-4)	2.8918(-4)	1.1486(-4)	4.8127(-4)	2.0048	1.9375	2.0154	1.9970
v_{xy}								
8	4.3915	4.5268(-1)	1.2230	2.7714				
16	1.8092	2.1577(-1)	6.5804(-1)	1.2441	1.2793	1.0690	0.8942	1.1556
32	7.5544(-1)	8.7502(-2)	3.2261(-1)	5.4792(-1)	1.2600	1.3021	1.0284	1.1830
64	3.3502(-1)	3.2714(-2)	1.5799(-1)	2.5132(-1)	1.1731	1.4194	1.0300	1.1244
128	1.5639(-1)	1.2605(-2)	7.8096(-2)	1.1967(-1)	1.0991	1.3759	1.0165	1.0705
v_{xx}								
8		2.5330(-1)	2.9069	4.7948				
16		1.2072(-1)	1.7689	2.7027		1.0692	0.7166	0.8271
32		5.1448(-2)	9.6294(-1)	1.4191		1.2304	0.8773	0.9294
64		2.2272(-2)	5.0114(-1)	7.2566(-1)		1.2079	0.94221	0.9677
128		1.0318(-2)	2.5552(-1)	3.6679(-1)		1.1100	0.9718	0.9843

Table 7.7: Test Problem 7.4.

N	$u(0.5,0.5)$	$M_x(1,0.5)$	$M_y(0.5,1)$
4	1.296998752664015(-3)	-4.885303342082698(-2)	-4.885303342082698(-2)
8	1.267796767803351(-3)	-5.132974343827812(-2)	-5.132974343827811(-2)
16	1.265486382533946(-3)	-5.133459829513500(-2)	-5.133459829513520(-2)
32	1.265329965618658(-3)	-5.133384158446099(-2)	-5.133384158446123(-2)
64	1.265319776660214(-3)	-5.133377004217192(-2)	-5.133377004217291(-2)
128	1.265319130702914(-3)	-5.133376508000668(-2)	-5.133376508000717(-2)

Table 7.8: Test Problem 7.5.

N	$u(0.5,0.5)$	$M_y(0.5,1)$
4	3.418370065472051(-3)	-9.656754445688147(-2)
8	4.769397134896294(-3)	-1.198296876081964(-1)
16	5.328637127458790(-3)	-1.244373503906979(-1)
32	5.523148208813368(-3)	-1.254491906133079(-1)
64	5.585311195584267(-3)	-1.256909975199914(-1)
128	5.604223281013911(-3)	-1.257507063440741(-1)

Chapter 8

CONCLUDING REMARKS

In this thesis, we have developed new QSC methods for solving Poisson's and biharmonic equations in the unit square, Ω. For Poisson's equation, we derived the new method from the work of Houstis et al., [20] and the work of Christara [13]. Then using a matrix decomposition algorithm with FFTs, we solved the resulting linear system on an $N \times N$ uniform partition of Ω at a cost of $O(N^2 \log N)$ operations. Next, we proved the existence and uniqueness of the QSC approximation, and then we presented numerical results that indicate global third-order accuracy and fourth-order accuracy at the nodal and collocation points and third-order accuracy at the Gauss points of the derivative approximation.

To solve the biharmonic Dirichlet problem, we extended our method for Poisson's equation after reducing the problem to a coupled system of two second-order partial differential equations in u and Δu. Based on the work of Christara [13], we perturbed the Neumann boundary conditions having shown the derivation with a fourth-order boundary-value problem. We then demonstrated how to solve the resulting linear system efficiently with the Schur complement approach and a matrix decomposition method which employs FFTs. The cost of the resulting algorithm is $O(N^3 \log N)$ for an $N \times N$ partition of Ω. Then, we present numerical results that indicate third-order global accuracy for the u and Δu approximations. Furthermore, we obtained fourth-order accuracy at the nodal and collocation points of the u approximation and third-order accuracy at the Gauss points of the approximations of the derivatives of u. For two of three test problems, we observed similar superconvergence results for the approximation of Δu.

The new QSC methods developed in this thesis are closely related to the MNCSC method developed by Abushama in [1], [2], and [3]. While less accurate globally than the MNCSC method, our new QSC method has several advantages. For example, the method can be implemented more simply since there are no corner equations. This means that in the quadratic case all coefficients are determined in a single linear system. Furthermore, the new QSC method presented here produces comparable accuracy at the nodal and collocation points. Compared to the work of Christara [13], the new methods here can each be solved efficiently with an MDA, which results in tridiagonal or banded systems of equations.

A few points for potential future research remain. The most notable area is proving the existence and uniqueness of the solution of the biharmonic Dirichlet problem, as well as a formal convergence analysis on the new method in each chapter.

The lack of superconvergence results for the approximation of Δu in solving the biharmonic Dirichlet problem in Problem 7.3 are unexplained as well as the lack of superconvergence in the approximation of the cross derivative of the Δu approximation in Problems 7.1 and 7.2. However, a similar reduction in accuracy was observed in [2].

Finally, the new QSC method could be extended to a more general domain than the unit square and to non-homogeneous boundary conditions.

REFERENCES CITED

[1] A. A. Abushama *Modified nodal cubic spline collocation for Poisson's and biharmonic equations in the unit square*, Ph.D. thesis, Colorado School of Mines, Golden, Colorado, 2004.

[2] A. A. Abushama and B. Bialecki, *Modified nodal cubic spline collocation for biharmonic equations*, Numer. Algorithms, 43 (2006), pp. 331–353.

[3] A. A. Abushama and B. Bialecki, *Modified nodal cubic spline collocation for Poisson's equation*, SIAM J. Numer. Anal., 46 (2008), pp. 397–418.

[4] M. Arad, A. Yakhot and G. Ben-Dor *A highly accurate numerical solution of a biharmonic equation*, Numer. Methods Partial Differential Eq., 13 (1997), pp. 375–391.

[5] D. Archer, *Some collocation methods for differential equations*, Ph.D. thesis, Rice University, Houston, Texas, 1973.

[6] D. Archer, *An $O(h^2)$ cubic spline collocation method for quasilinear parabolic equations*, SIAM J. Numer. Anal., 14 (1977), pp. 620–637.

[7] B. Bialecki, *A fast solver for the orthogonal spline collocation solution of the biharmonic Dirichlet problem on rectangles*, J. Comput. Phys., 191 (2003), pp. 601–621.

[8] B. Bialecki, G. Fairweather, and A. Karageorghis, *Matrix decomposition algorithms for modified spline collocation for Helmholtz problems*, SIAM J. Sci. Comput., 24 (2003), 1733–1753.

[9] B. Bialecki, G. Fairweather, and A. Karageorghis, *Optimal superconvergent one step nodal cubic spline collocation methods*, SIAM J. Sci. Comput., 27 (2005), 575–598.

[10] B. Bialecki, G. Fairweather, A. Karageorghis and Q. Nguyen, *Optimal superconvergent one step quadratic spline collocation methods*, BIT, 48 (2008), pp. 449–472.

[11] B. Bialecki, G. Fairweather, A. Karageorghis and Q. Nguyen, *Optimal superconvergent one step quadratic spline collocation methods for Helmholtz equations*, in Recent Advances in Computational Science, P. Jorgensen, X. Shen, C–W. Shu and N. Yan, eds., World Scientific, Singapore, 2008, pp. 156-174.

[12] P. Bjørstad, *Fast numerical solution of the biharmonic Dirichlet problem on rectangles*, SIAM J. Numer. Anal., 20 (1983), pp. 59–71.

[13] C. C. Christara, *Quadratic spline collocation methods for elliptic partial differential equations*, BIT, 34 (1994), pp. 33–61.

[14] C. de Boor, *The method of projections as applied to the numerical solution of two point boundary value problems using cubic splines*, Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, 1966.

[15] C. de Boor and B. Swartz, *Collocation at Gaussian points*, SIAM J. Numer. Anal., 10 (1973), pp. 582–606.

[16] J. W. Daniel and B. K. Swartz, *Extrapolated collocation for two–point boundary value problems using cubic splines*, J. Inst. Math. Appl., 16 (1975), pp. 161–174.

[17] L. W. Ehrlich, *Solving the biharmonic equation as coupled finite difference equations*, SIAM J. Numer. Anal. 8 (1971), pp 278–287.

[18] D. J. Fyfe, *The use of cubic splines in the solution of two–point boundary value problems*, Comput. J., 13 (1969), pp. 188-192.

[19] W. Hackbusch, *Elliptic Differential Equations, Theory and Numerical Treatment*, Springer Verlag, New York, 1992.

[20] E. N. Houstis, C. C. Christara and J. R. Rice, *Quadratic-spline collocation methods for two-point boundary value problems*, Internat. J. Numer. Methods Engrg., 26 (1988), pp. 935–952.

[21] E. N. Houstis, E. A. Vavalis and J. R. Rice, *Convergence of $O(h^4)$ cubic spline collocaton methods for elliptic partial differential equation*, SIAM J. Numer. Anal., 25 (1988), pp. 54–74.

[22] D. B. Knudson, *A piecewise Hermite bicubic finite element Galerkin method for the biharmonic Dirichlet problem*, Ph.D. thesis, Colorado School of Mines, Golden, Colorado, 1997.

[23] V. V. Meleshko, *Biharmonic problem in a rectangle*, Applied Scientific Research, 58 (1998), pp. 217–249.

[24] Q. N. Nguyen, *Matrix decomposition algorithms for modified quadratic spline collocation for Helmholtz problems*, M.S. thesis, Colorado School of Mines, Golden, Colorado, 2005.

[25] P. Peiker, *On the numerical solution of the first biharmonic equation*, Model. Math. Anal. Numer., 22 (1988), pp. 655–676.

[26] X. Zhang, *Multilevel Schwarz methods for the biharmonic Dirichlet problem*, SIAM J. Sci. Comput., 15 (1994), pp. 621–644.