1984

Tensor Product Generalized ADI Methods for Elliptic Problems

Wayne R. Dyksen

Report Number:

84-493

TENSOR PRODUCT GENERALIZED ADI METHODS FOR ELLIPTIC PROBLEMS

Wayne R. Dyksen

Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907

CSD-TR 493
September 1984

ABSTRACT

We consider solving separable, second order, linear elliptic partial differential equations. If an elliptic problem is separable, then, for certain discretizations, the matrices involved in the corresponding discrete problem can be expressed in terms of tensor products of lower order matrices. In the most general case, the discrete problem can be written in the form $(A_1 \otimes B_2 + B_1 \otimes A_2)C = F$. We present a new Tensor Product Generalized Alternating Direction Implicit (TPGADI) iterative method for solving such discrete problems. We prove convergence and establish computational efficiency. The TPGADI method is applied to the Hermite bicubic collocation equations. We conclude that the TPGADI method is an effective tool for solving the discrete elliptic problems arising from a large class of elliptic problems.

Tensor Product Generalized ADI Methods for Elliptic Problems

Wayne R. Dyksen

1. Introduction

We present new methods for solving the discrete problems arising from separable, second order, linear elliptic partial differential equations. The methods we present are natural products of the classical approach. If a problem is separable, then its solution can be expressed in terms of tensor products of solutions of lower dimensional problems, and hence is reduced to that of solving much simpler problems. For certain discretizations, this means that the matrices involved in the corresponding discrete problem can be expressed in terms of tensor products of lower order matrices. For example, in the most general case the system of linear equations which constitute the discrete problem can be written as $(A_1 \otimes B_2 + B_1 \otimes A_2)C = F$.

We begin in Section 2 with a brief introduction to some theoretical and computational aspects of tensor products and matrices. In Section 3 we present a new method which we call the Tensor Product Generalized Alternating Direction Implicit (TPGADI) method for solving discrete elliptic problems of the form $(A_1 \otimes B_2 + B_1 \otimes A_2)C = F$. In Sections 4, 5 and 6 we apply the TPGADI method to solve the Hermite bicubic collocation equations. We show that the TPGADI method is an effective tool for solving the discrete elliptic problems arising from a large class of elliptic problems. In Section 7 we summarize our results.

2. Tensor Products of Matrices

Let $A = \{a_{mn}\}$ and $B = \{b_{kl}\}$ be matrices of order $M \times N$ and $K \times L$, respectively. The **tensor product (Kronecker product, direct product)** of A and B, denoted by $A \otimes B$, is the matrix of order $MK \times NL$ given by

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1N}B \\ a_{21}B & a_{22}B & \dots & a_{2N}B \\ \vdots & \vdots & & \vdots \\ a_{M1}B & a_{M2}B & \dots & a_{MN}B \end{bmatrix}.$$

Some of the properties of tensor products are summarized below; a detailed account is given in [Halmos, 1958].

$$(A_1 + A_2) \otimes B = A_1 \otimes B + A_2 \otimes B$$

$$A \otimes (B_1 + B_2) = A \otimes B_1 + A \otimes B_2$$

$$(A_1 \otimes A_2)(B_1 \otimes B_2) = A_1 B_1 \otimes A_2 B_2$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$(A \otimes B)^T = A^T \otimes B^T .$$

Note that if $\mathbf{x}$ and $\mathbf{y}$ are eigenvectors of A and B with eigenvalues λ and μ, respectively, then $\mathbf{x} \otimes \mathbf{y}$ is an eigenvector of $A \otimes B$ with corresponding eigenvalue $\lambda\mu$.

The fact that a particular matrix factors into the tensor product of two or more matrices is of no value without algorithms for doing efficient computer manipulation of tensor products. For example, to compute $(A \otimes B)\mathbf{x}$ we must use only the factors A and B, and avoid explicitly forming the tensor product $A \otimes B$. Such algorithms are given in [de Boor, 1979].

When computing with tensor products, it is computationally convenient to represent vectors using matrices. For example, when working with $(A \otimes B)\mathbf{x}$, we represent the NL-vector $\mathbf{x}$ by the matrix $X = \{x_{ln}\}$ of order $L \times N$ defined by

$$x_{ln} = \mathbf{x}_{l+L(n-1)}.$$

The usefulness of this representation can be seen in the following simple results which give efficient procedures for computing $(A \otimes B)\mathbf{x}$ and solving $(A_1 \otimes A_2)\mathbf{x} = \mathbf{b}$, respectively.

Lemma 2.1. *Let $A = \{a_{mn}\}$, $B = \{b_{kl}\}$ and $X = \{x_{ln}\}$ be matrices of order $M \times N$, $K \times L$ and $L \times N$, respectively. Then the $K \times M$ matrix $(A \otimes B)X$ is given by*

$$(A \otimes B)X = (A(BX)^T)^T.$$

Corollary 2.2. *Let A_k be matrices of order $N_k \times N_k$, let X and B be matrices of order $N_2 \times N_1$, and consider the linear system*

$$(A_1 \otimes A_2)X = B.$$

If A_1^{-1} and A_2^{-1} exist, and if $A_2Y = B$ and $A_1Z = Y^T$, then $X = Z^T$.

Since we make extensive use of these two basic tensor product operations in the case in which the factors are band matrices, we give here their computational complexity. Let A_k be matrices of order $N_k \times N_k$ with bandwidth K_k, and let B and X be matrices of order $N_2 \times N_1$. Then the work to compute $(A_1 \otimes A_2)X$ is easily computed to be $O(2N_1N_2(K_1+K_2))$. The work to solve $(A_1 \otimes A_2)X = B$ using Gauss elimination with partial pivoting is given in Table 2.1 which shows that the work is $O(2K_1^2N_1 + 2K_2^2N_2 + 3N_1N_2(K_1+K_2))$.

Table 2.1
Work to solve $(A_1 \otimes A_2)X = B$

Operation	Work
Factor A_2	$2K_2^2N_2$
Solve $L_2U_2Y = B$	$3N_1K_2N_2$
Factor A_1	$2K_1^2N_1$
Solve $L_1U_1Z = Y^T$	$3K_1N_1N_2$

Observe that dominant work results from handling the multiple right sides B and Y^T since $K_k << N_k$. On a computer which provides the facility for doing parallel processing, the forward and back substitutions can be done simultaneously for all right sides, reducing the work by approximately an order of magnitude to $O(3(K_1N_1+K_2N_2))$.

The need to use band Gauss elimination with partial pivoting to solve $A_2Y=B$ and $A_1Z=Y^T$ is, in some sense, a worst case. In particular, we may want to solve $(A_1\otimes I)X=B$ or perhaps $(A_1\otimes A_2)X=B$ where A_2 is symmetric, positive definite so that the work estimates given above are indeed over estimates. In many applications, A_1 and/or A_2 have nice properties which $A_1\otimes A_2$ does not share.

The linear systems arising from separable elliptic problems do not factor into the simple form $(A_1\otimes A_2)\mathbf{x}=\mathbf{b}$; instead, they are of the form $(A_1\otimes B_2+B_1\otimes A_2)\mathbf{x}=\mathbf{b}$. The simple procedures considered here are employed to solve such equations.

3. The Two Directional Tensor Product Generalized ADI Methods

Let A_k and B_k be matrices of order $N_k\times N_k$, and consider the linear system

$$(A_1\otimes B_2+B_1\otimes A_2)C=F. \tag{3.1}$$

While the tensor product $(A_1\otimes B_2+B_1\otimes A_2)$ is an $N_1N_2\times N_1N_2$ matrix, we wish to solve (3.1) by computing only with A_1, B_1 and A_2, B_2; that is, we wish to solve the two directional problem (3.1) by using methods employed to solve the one directional problems. We use the term *directional* rather than *dimensional* since one direction may encompass more than one dimension, as in the Method of Planes [Dyksen, 1984d].

For a given set of positive *acceleration parameters* ρ_k, $k=1,2,\ldots$, we define the two directional *Tensor Product Generalized Alternating Direction Implicit* (TPGADI) iteration method by

$$\begin{aligned} &C^{(0)} \text{ given} \\ &\Big[(A_1+\rho_{k+1}B_1)\otimes B_2\Big]C^{(k+\frac{1}{2})}=F-\Big[B_1\otimes(A_2-\rho_{k+1}B_2)\Big]C^{(k)} \\ &\Big[B_1\otimes(A_2+\rho_{k+1}B_2)\Big]C^{(k+1)}=F-\Big[(A_1-\rho_{k+1}B_1)\otimes B_2\Big]C^{(k+\frac{1}{2})}. \end{aligned} \tag{3.2}$$

The TPGADI method is a natural extension of the standard Peaceman-Rachford ADI method [Peaceman and Rachford, 1955]. In fact, with $B_k = I_k$, the identity matrix of order k, (3.2) reduces to the tensor product ADI schemes presented in [Lynch, Rice and Thomas, 1964a, 1964b, 1965].

THEOREM 3.1. *Let A_k and B_k be matrices of order $N_k \times N_k$, and consider the linear system* (3.1) *for F given. Suppose that $B_1^{-1}A_1$ and $B_2^{-1}A_2$ have complete sets of normalized eigenvectors p_i and q_j, respectively, with corresponding positive eigenvalues λ_i and μ_j, respectively. Then, for a given set of positive acceleration parameters ρ_k, $k=1,2,\ldots$, the two directional Tensor Product Generalized Alternating Direction Implicit iterative method, given by* (3.2) *is convergent, and C is its only solution.*

Proof. Let $E^{(k)} = C^{(k)} - C$ denote the error of the k^{th} iterate, and let I_k denote the identity matrix of order k. A straightforward computation shows that the error satisfies

$$E^{(0)} = C^{(0)} - C$$

(3.3)
$$E^{(k+1)} = \Big[(B_1^{-1}A_1 - \rho_{k+1}I_1)(B_1^{-1}A_1 + \rho_{k+1}I_1)^{-1}$$
$$\otimes (B_2^{-1}A_2 + \rho_{k+1}I_2)^{-1}(B_2^{-1}A_2 - \rho_{k+1}I_2)\Big]E^{(k)}.$$

If we expand the error $E^{(k)}$ in terms of the eigenvectors of $B_1^{-1}A_1$ and $B_2^{-1}A_2$ as

(3.4)
$$E^{(k)} = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} E_{ij}^{(k)} p_i \otimes q_j$$

and substitute (3.4) into (3.3), we obtain

$$E^{(k+1)} = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\left[\frac{\lambda_i - \rho_{k+1}}{\lambda_i + \rho_{k+1}}\;\frac{\mu_j - \rho_{k+1}}{\mu_j + \rho_{k+1}}\right] E_{ij}^{(k)} p_i \otimes q_j .$$

Hence, the error $E^{(k)}$ may be expressed in terms of the initial error $E^{(0)}$ as

$$E^{(k)} = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \prod_{l=1}^{k} \left[\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l} \; \frac{\mu_j - \rho_l}{\mu_j + \rho_l} \right] E_{ij}^{(0)} p_i \otimes q_j$$

so that

$$E_{ij}^{(k)} = \prod_{l=1}^{k} \left[\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l} \; \frac{\mu_j - \rho_l}{\mu_j + \rho_l} \right] E_{ij}^{(0)}. \tag{3.5}$$

Since by the hypothesis the eigenvalues λ_i and μ_j are positive, it follows from (3.5) that for positive acceleration parameters ρ_l

$$\lim_{k \to \infty} |E_{ij}^{(k)}| = \lim_{k \to \infty} \prod_{l=1}^{k} \left| \frac{\lambda_i - \rho_l}{\lambda_i + \rho_l} \; \frac{\mu_j - \rho_l}{\mu_j + \rho_l} \; E_{ij}^{(0)} \right| = 0,$$

so that

$$\lim_{k \to \infty} \| E^{(k)} \| = 0$$

which is the desired result □

We see from (3.5) that $E_{ij}^{(k)}$ can be made zero for all j by taking $\rho_l = \lambda_i$ for some i. This observation makes transparent the power of the TPGADI method, namely, that many (N_1 or N_2) components of the error vector can be annihilated at the same time. Moreover, if the λ_i, μ_j and ρ_l are positive, then this annihilation is accomplished without simultaneously magnifying any other components of the error.

Corollary 3.2. *The TPGADI iterative method* (3.2) *can be exact* (*except for round-off*) *in a number of iterations equal to the number of unknowns in either direction; that is, in N_1 or N_2 iterations.*

Proof. Let $\lambda_1, \ldots, \lambda_{N_1}$ be the eigenvalues of $B_1^{-1} A_1$ and set $\rho_l = \lambda_l$. Then by (3.5) we have for all i

$$E_{ij}^{(N_1)} = \prod_{l=1}^{N_1} \left[\frac{\lambda_i - \rho_l}{\lambda_i + \rho_l} \; \frac{\mu_j - \rho_l}{\mu_j + \rho_l} \right] E_{ij}^{(0)} = 0.$$

Thus,

$$E^{(N_1)} = 0.$$

The analogous argument for N_2 iterations completes the proof □

The TPGADI method (3.2) is one member of a general family of TPGADI methods defined by

$$\begin{aligned}
&C^{(0)} \text{ given} \\
&\left[(A_1 + \rho_{k+1} B_1) \otimes B_2\right] C^{(k+1/2)} = F - \left[B_1 \otimes (A_2 - \rho_{k+1} B_2)\right] C^{(k)} \\
&\left[B_1 \otimes (A_2 + \rho_{k+1} B_2)\right] C^{(k+1)} = \left[B_1 \otimes (A_2 - \omega\rho_{k+1} B_2)\right] C^{(k)} \\
&\qquad\qquad + (1+\omega)\rho_{k+1}(B_1 \otimes B_2) C^{(k+1/2)}
\end{aligned}$$

where ω is a fixed scalar and ρ_k are positive acceleration parameters. The values $\omega = 1, 0$ correspond to generalizations of the Peaceman-Rachford method and the Douglas-Rachford method, respectively [Douglas and Rachford, 1956].

To compare the TPGADI method to other schemes, we estimate the computer time (via operation counts) and computer memory required to implement it. We assume that A_k and B_k are band matrices with bandwidth K_k and that all systems of linear equations are solved by Gauss elimination with partial pivoting. Since the initial guess $C^{(0)}$ and the acceleration parameters ρ_k depend on the discretization method used, we assume here that they are given.

The work to compute the 1-direction sweep of the TPGADI method (3.2) is estimated using the results of Table 2.1. We obtain the following:

Table 3.1
Work to compute the 1-direction sweep of the TPGADI method

Operation	Work
$W_2 = A_2 - \rho_{k+1}B_2$	$2K_2N_2$
$W = (B_1 \otimes W_2)C^{(k)}$	$2N_1N_2(K_1+K_2)$
$W = F - W$	$\frac{1}{2}N_1N_2$
$W_1 = A_1 + \rho_{k+1}B_1$	$2K_1N_1$
$C^{(k+\frac{1}{2})} = (W_1 \otimes B_2)^{-1}W$	$2K_1^2N_1 + 2K_2^2N_2 + 3N_1N_2(K_1+K_2)$

Thus, the total work to compute the 1-direction sweep is $O\left(N_1N_2(5(K_1+K_2)+\frac{1}{2})\right)$. An analogous estimate shows that the work for the 2-direction sweep is the same. Hence, the total work per iteration is $O\left(N_1N_2(10(K_1+K_2)+1)\right)$ operations. If $N_1=N_2=N$ and $K_1=K_2=K$, then this work estimate simplifies to $O(20KN^2)$.

Note that the dominant work in Table 3.1 does not result from factoring W_1 or B_2. Instead, the dominant work involves computing the right side W and doing multiple back substitutions solving for $C^{(k+\frac{1}{2})}$, operations which are often negligible in other applications. On a computer with parallel computing facilities, a large gain in speed could result by doing the multiple back substitutions in parallel.

We now compare the TPGADI method to the straight forward method of applying simple band Gauss elimination to the matrix $A = (A_1 \otimes B_2 + B_1 \otimes A_2)$. If $N_1=N_2=N$ and $K_1=K_2=K$, then the matrix A is of order $N^2 \times N^2$ with approximate bandwidth KN so that band Gauss elimination with partial pivoting applied to it requires $O(2K^2N^4)$ operations. The TPGADI iterative method can be a direct method in N iterations, requiring $O(20KN^3)$ operations. Thus, the TPGADI method is asymptotically much faster than straight forward Gauss elimination as a direct method of solution.

The analysis warrants a few remarks. First, in order for the TPGADI method to be direct we must either know *a priori* the N eigenvalues of $B_1^{-1}A_1$ or $B_2^{-1}A_2$ or must compute them; in the applications we consider, the computation of these eigenvalues is insignificant. Second, given the desired eigenvalues, we could use some subset of them to achieve moderate accuracy with many fewer than N iterations; we discuss this in Section 6.

A simple calculation shows that the amount of memory required to factor the matrix $(A_1 \otimes B_2 + B_1 \otimes A_2)$ by Gauss elimination with partial pivoting is $O(3KN^3 + 2N^2)$ words. The memory requirements for the TPGADI method are estimated as follows: A_1, B_1, A_2, B_2 each require $O(2KN)$ words; W_1 and W_2 require $O(3KN)$ words; and W, F and C each require N^2 words. Thus, the total amount of computer memory required is $O(3N^2)$ words, which is nearly optimal since it is the same order of magnitude as the number N^2 of unknowns.

4. Collocation with Hermite Bicubics

We consider an elliptic problem of the form

$$\begin{aligned} L_x u + L_y u &= f \quad \text{in } \Omega = [0,1] \times [0,1] \\ u &= 0 \quad \text{on } \partial\Omega , \end{aligned} \tag{4.1}$$

where

$$L_x u = -a_2(x) u_{xx} + a_1(x) u_x + a_0(x) u , \; a_2 > 0,$$

$$L_y u = -b_2(y) u_{yy} + b_1(y) u_y + b_0(y) u , \; b_2 > 0.$$

We assume for simplicity that we have homogeneous Dirichlet boundary conditions. The analysis is readily extended to problems with nonhomogeneous Dirichlet or Neumann boundary conditions [Dyksen, 1984d], [Houstis, et. al., 1983a, 1983b].

The domain Ω is subdivided with a rectangular, tensor product grid with MN rectangles. We approximate $u(x,y)$ by

$$U(x,y)=\sum_{m=1}^{2M}\sum_{n=1}^{2N}c_{nm}\phi_m(x)\psi_n(y)$$

where ϕ_m and ψ_n are the standard one dimensional Hermite cubics with the grid lines as knots. The Hermite cubics which are zero on $\partial\Omega$ are discarded so that $U=0$ on $\partial\Omega$.

To determine the $4MN$ unknowns c_{nm}, we place in each subinterval (x_m, x_{m+1}) and (y_n, y_{n+1}), the two Gauss points $\tau_{2m+1}=½(x_m+x_{m+1})-\frac{h_x}{2\sqrt{3}}$, $\tau_{2m+2}=½(x_m+x_{m+1})+\frac{h_x}{2\sqrt{3}}$ and $\upsilon_{2n+1}=½(y_n+y_{n+1})-\frac{h_y}{2\sqrt{3}}$, $\upsilon_{2n+2}=½(y_n+y_{n+1})+\frac{h_y}{2\sqrt{3}}$. These collocation points give a fourth order discretization error for smooth problems [Houstis, 1978], [Percell and Wheeler, 1980]. We then collocate the elliptic problem (4.1) at these $4MN$ points to obtain the *Hermite bicubic collocation equations*

$$(4.2)\qquad L_x[U](\tau_i,\upsilon_j)+L_y[U](\tau_i,\upsilon_j)=f(\tau_i,\upsilon_j)\quad \begin{array}{l} i=1,\ldots,2M \\ j=1,\ldots,2N. \end{array}$$

The structure of the linear system in (4.2) depends on the ordering of the collocation points and the basis functions [Rice, 1981a]. If they are both ordered in a natural tensor product manner, then (4.2) may be written in tensor product form as

$$(A_x\otimes B_y+B_x\otimes A_y)C=F,$$

where

$$[A_x]_{im}=L_x\phi_m(\tau_i),\quad [B_x]_{im}=\phi_m(\tau_i),\quad \begin{array}{l} i=1,\ldots,2M \\ m=1,\ldots,2M, \end{array}$$

$$[A_y]_{jn}=L_y\psi_n(\upsilon_j),\quad [B_y]_{jn}=\psi_n(\upsilon_j),\quad \begin{array}{l} j=1,\ldots,2N \\ n=1,\ldots,2N, \end{array}$$

$$C_{nm}=c_{nm},\ \begin{array}{l} n=1,\ldots,2N \\ m=1,\ldots,2M, \end{array}\quad \text{and}\quad F_{ji}=f(\tau_i,\upsilon_j),\ \begin{array}{l} j=1,\ldots,2N \\ i=1,\ldots,2M. \end{array}$$

Since the support of each Hermite cubic ϕ_m and ψ_n spans at most two subintervals, it follows that A_x, B_x and A_y, B_y have bandwidth two, regardless of M or N.

5. The TPGADI Method Applied to the Collocation Equations

We now apply the TPGADI method (3.2) to the Hermite bicubic collocation equations. In particular, we establish the convergence of the TPGADI method when applied to the *Discrete Model Problem* arising from the *Model Problem*

$$\begin{aligned} -u_{xx} - u_{yy} &= f \quad \text{in } \Omega = [0,1]\times[0,1] \\ u &= 0 \quad \text{on } \partial\Omega . \end{aligned} \tag{5.1}$$

For the Discrete Model Problem, the matrices A_x, B_x and A_y, B_y in (3.3) are defined by $[A_x]_{lm} = -\phi_m''(\tau_l)$, $[B_x]_{lm} = \phi_m(\tau_l)$, and $[A_y]_{jn} = -\psi_n''(\upsilon_j)$, $[B_y]_{jn} = \psi_n(\upsilon_j)$.

Since convergence of the TPGADI method depends on generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ and $A_y\mathbf{c} = \lambda B_y\mathbf{c}$, we consider the classical eigenvalue problem

$$\begin{aligned} &u''(x) = \lambda u(x), \quad x \in (0,1) \\ &u(0) = u(1) = 0. \end{aligned} \tag{5.2}$$

We divide the unit interval into N equal subintervals of length $h = 1/N$. We approximate an eigenfunction u of (5.2) by $U(x) = \sum_{l=1}^{2N} c_l\phi_l(x)$ for some constants c_l, where the ϕ_l are the $2N$ Hermite cubics associated with the $N+1$ grid points $x_k = kh$, and which satisfy $\phi_l(0) = \phi_l(1) = 0$. For a fixed parameter $0 < \theta < ½$, we place in each subinterval (x_k, x_{k+1}) two collocation points, $\tau_{2k+1} = ½(x_k + x_{k+1}) - \theta h$ and $\tau_{2k+2} = ½(x_k + x_{k+1}) + \theta h$. Substituting U into (5.2) and collocating at these points, we obtain the generalized eigenvalue problem

$$A\mathbf{c} = \lambda B\mathbf{c}, \tag{5.3}$$

where

$$A_{lj} = \phi_j''(\tau_l), \quad B_{lj} = \phi_j(\tau_l), \quad \begin{matrix} l = 1,\ldots,2N \\ j = 1,\ldots,2N. \end{matrix}$$

The generalized eigenvalues and eigenvectors of (5.3) give the Hermite collocation approximations to the eigenvalues and eigenvectors of (5.2).

THEOREM 5.1. *The 2N generalized eigenvalues of* $A\mathbf{c} = \lambda B \mathbf{c}$ *in* (5.3) *are given by*

$$\lambda_0 = \frac{6}{h^2(\theta^2 - 1/4)} \tag{5.4a}$$

$$\lambda_N = \frac{2}{h^2(\theta^2 - 1/4)} \tag{5.4b}$$

$$\lambda_l^{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \ , \ l = 1,\ldots,N-1 \tag{5.4c}$$

where

$$a = h^4\Big[(16\theta^4 - 16\theta^2 + 3)d \ - 8\theta^2 + 2\Big], \tag{5.5a}$$

$$b = h^2\Big[(-128\theta^2 + 48)d \ + 48\Big], \tag{5.5b}$$

$$c = 192d \ , \tag{5.5c}$$

and where

$$d = \tan^2(lh\pi/2). \tag{5.5d}$$

Proof. Let P be the Hermite cubic collocation approximation of the eigenfunction of (5.2) corresponding to the approximate eigenvalue λ. Since $h = 1/N$, P consists of N pieces, $p_k(x)$, each of which has support in (x_k, x_{k+1}), $k = 0,\ldots,N-1$. Denoting the kth piece of P'' by $p_k''(x) = \gamma_k + \delta_k x$, we have

$$p_k(x) = \alpha_k + \beta_k x + \gamma_k \frac{x^2}{2} \ + \delta_k \frac{x^3}{6} \ . \tag{5.6}$$

We assume for the sake of simplicity that each polynomial piece is centered at the midpoint of its corresponding interval.

First, we relate the α_k's to the γ_k's and the β_k's to the δ_k's by using the eigenvalue problem. Since P satisfies $P'' = \lambda P$ at the collocation points, we have $p_k''(\pm\theta h) = \lambda p_k(\pm\theta h)$, or equivalently,

$$\gamma_k \pm \delta_k \theta h = \lambda \left(\alpha_k \pm \beta_k \theta h + \gamma_k \frac{\theta^2 h^2}{2} \pm \delta_k \frac{\theta^3 h^3}{6} \right). \tag{5.7}$$

Adding and subtracting the equations in (5.7), we obtain, respectively,

$$\gamma_k = \lambda \left(\alpha_k + \gamma_k \frac{\theta^2 h^2}{2} \right)$$
$$\delta_k = \lambda \left(\beta_k + \delta_k \frac{\theta^2 h^2}{6} \right),$$

from which we have

$$\alpha_k = C_\alpha \gamma_k = \left(\frac{1}{\lambda} - \frac{\theta^2 h^2}{2} \right) \gamma_k$$
$$\beta_k = C_\beta \delta_k = \left(\frac{1}{\lambda} - \frac{\theta^2 h^2}{6} \right) \delta_k .$$

Thus, (5.6) simplifies to

$$p_k(x) = \left(C_\alpha + \frac{x^2}{2} \right) \gamma_k + \left(C_\beta x + \frac{x^3}{6} \right) \delta_k .$$

Next, we relate the γ_k's to the δ_k's by using the continuity of P and P'. Since P is continuous, we have $p_k(h/2) = p_{k+1}(-h/2)$, or equivalently,

$$\left(C_\alpha + \frac{h^2}{8} \right) \gamma_k + \left(C_\beta \frac{h}{2} + \frac{h^3}{48} \right) \delta_k = \left(C_\alpha + \frac{h^2}{8} \right) \gamma_{k+1} - \left(C_\beta \frac{h}{2} + \frac{h^3}{48} \right) \delta_{k+1},$$

which we write as

$$r(-\gamma_k + \gamma_{k+1}) = s(\delta_k + \delta_{k+1}), \tag{5.8}$$

where $r = C_\alpha + \frac{h^2}{8}$ and $s = C_\beta \frac{h}{2} + \frac{h^3}{48}$. Furthermore, since P' is continuous, we have $p_k'(h/2) = p_{k+1}'(-h/2)$, or equivalently,

$$\frac{h}{2} \gamma_k + \left(C_\beta + \frac{h^2}{8} \right) \delta_k = -\frac{h}{2} \gamma_{k+1} + \left(C_\beta + \frac{h^2}{8} \right) \delta_{k+1},$$

which we write as

$$\gamma_k + \gamma_{k+1} = t(-\delta_k + \delta_{k+1}), \qquad t = \frac{2}{h}\left(C_\beta + \frac{h^2}{8}\right) \tag{5.9}$$

Now, using (5.8) and (5.9), we show that the γ_k's and δ_k's both satisfy the same difference equation. We consider (5.8) and the equation obtained from it by replacing k by $k-1$. We obtain

$$\begin{aligned} r(-\gamma_k + \gamma_{k+1}) &= s(\delta_k + \delta_{k+1}) \\ r(-\gamma_{k-1} + \gamma_k) &= s(\delta_{k-1} + \delta_k), \end{aligned} \tag{5.10}$$

which, if added, yield

$$r(-\gamma_{k-1} + \gamma_{k+1}) = s(\delta_{k-1} + 2\delta_k + \delta_{k+1}). \tag{5.11}$$

Similarly, from (5.9) we obtain

$$\begin{aligned} \gamma_k + \gamma_{k+1} &= t(-\delta_k + \delta_{k+1}) \\ \gamma_{k-1} + \gamma_k &= t(-\delta_{k-1} + \delta_k), \end{aligned} \tag{5.12}$$

which, if subtracted, yield

$$\gamma_{k-1} - \gamma_{k+1} = -t(\delta_{k-1} - 2\delta_k + \delta_{k+1}). \tag{5.13}$$

Substituting (5.13) into (5.11) gives

$$rt(\delta_{k-1} - 2\delta_k + \delta_{k+1}) = s(\delta_{k-1} + 2\delta_k + \delta_{k+1}). \tag{5.14}$$

If we subtract the equations in (5.10) and add the equations in (5.12), we obtain, respectively,

$$\begin{aligned} r(\gamma_{k-1} - 2\gamma_k + \gamma_{k+1}) &= s(-\delta_{k-1} + \delta_{k+1}) \\ \gamma_{k-1} + 2\gamma_k + \gamma_{k+1} &= t(-\delta_{k-1} + \delta_{k+1}), \end{aligned}$$

which gives

$$rt(\gamma_{k-1} - 2\gamma_k + \gamma_{k+1}) = s(\gamma_{k-1} + 2\gamma_k + \gamma_{k+1}). \tag{5.15}$$

Now, since the γ_k's and δ_k's satisfy the difference equations in (5.14) and (5.15), we may in the usual way set

$$\gamma_k = A_k \zeta^k + C_k \zeta^{-k}$$
$$\delta_k = B_k \zeta^k + D_k \zeta^{-k}.$$

However, the eigenvalue problem is invariant with respect to translation; that is, we must have $\gamma_0 = \pm \gamma_{N-1}$ and $\delta_0 = \pm \delta_{N-1}$. Thus we may set

(5.16)
$$\gamma_k = A_k \sin\left(\frac{(k+½)l\pi}{N}\right) + C_k \cos\left(\frac{(k+½)l\pi}{N}\right)$$
$$\delta_k = B_k \sin\left(\frac{(k+½)l\pi}{N}\right) + D_k \cos\left(\frac{(k+½)l\pi}{N}\right).$$

Substituting (5.16) into (5.14) and (5.15), and simplifying, we obtain

(5.17)
$$rt\left(-4\sin^2\left(\frac{l\pi}{2N}\right)\delta_k\right) = s\left(4\cos^2\left(\frac{l\pi}{2N}\right)\delta_k\right)$$
$$rt\left(-4\sin^2\left(\frac{l\pi}{2N}\right)\gamma_k\right) = s\left(4\cos^2\left(\frac{l\pi}{2N}\right)\gamma_k\right).$$

Since r, s, and t depend on λ, it follows from (5.17) that the eigenvalues of (5.3) satisfy

(5.18)
$$rt\sin^2\left(\frac{l\pi}{2N}\right) + s\cos^2\left(\frac{l\pi}{2N}\right) = 0.$$

We can now obtain the formulas given in (5.4) by considering (5.18) for various values of l. If $l = 0$, then (5.18) reduces to

$$s = \left[\frac{1}{\lambda} - \frac{\theta^2 h^2}{6}\right]\frac{h}{2} + \frac{h^3}{48} = 0,$$

or equivalently,

(5.19)
$$\lambda_0 = \frac{6}{h^2(\theta^2 - ¼)},$$

which is (5.4a). Note that (5.19) may also be written as

$$6(\pm\,\theta h) = \lambda(\pm\,\theta h)(\theta^2 h^2 - h^2/4),$$

which shows that the approximate eigenfunction associated with λ_0 is given up to a multiplicative constant by $p_k(\lambda_0;x) = x(x^2 - h^2/4)$. Moreover, $p_k(\lambda_0;x)$ satisfies the boundary conditions, $p_k(\lambda_0;\pm h/2) = 0$, and is a piecewise approximation to the eigenfunction $\sin(2N\pi x)$ of (5.2).

If $l = N$, then (5.18) implies $rs = 0$ so that either

(5.20)
$$r = \frac{1}{\lambda} - \frac{\theta^2 h^2}{2} + \frac{h^2}{8} = 0$$

or

(5.21)
$$s = \frac{2}{h}\left(\frac{1}{\lambda} - \frac{\theta^2 h^2}{6} + \frac{h^2}{8}\right) = 0.$$

From (5.20) we obtain (5.4b),

$$\lambda_N = \frac{2}{h^2(\theta^2 - 1/4)}\ .$$

The approximate eigenfunction corresponding to λ_N is given up to a multiplicative constant by $p_k(\lambda_N;x) = \pm\,(x^2 - h^2/4)$. Note that $p_k(\lambda_N;x)$ satisfies the boundary conditions, $p_k(\lambda_N;\pm h/2) = 0$, and is a piecewise approximation to the eigenfunction $\sin(N\pi x)$ of (5.2).

From (5.21) it follows that $\bar{\lambda}_N = \dfrac{6}{h^2(\theta^2 - 3/4)}$ with corresponding approximate eigenfunction $p_k(\bar{\lambda}_N;x) = x\left(x^2 - \dfrac{3h^2}{4}\right)$. Since $p_k(\bar{\lambda}_N;\pm h/2) \neq 0$, $\bar{\lambda}_N$ is not an eigenvalue of (5.3).

Finally, for $l = 1,\ldots,N-1$, we have from (5.18) that

(5.22)
$$\lambda^2 rs \tan^2\left(\frac{l\pi}{2N}\right) + \lambda^2 s = 0$$

which is a quadratic equation in λ. If simplified, (5.22) may be written as

(5.23)
$$a\lambda^2 + b\lambda + c = 0,$$

where a, b, c and d are given in (5.5). Thus, for each of $l = 1, \ldots, N-1$, (5.23) represents two eigenvalues of (5.3) which gives (5.5c),

$$\lambda_l^{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \Box$$

By varying the free parameter $0 < \theta < ½$ in Theorem 5.1, we can vary the location of the $2N$ collocation points τ_l, thereby affecting the accuracy of the approximations to the eigenvalues of (5.2).

Corollary 5.2. *If $0 < \theta < ½$, then λ_1^+ is at least an $O(h^2)$ approximation to the eigenvalue of smallest magnitude of (5.2), $-\pi^2$. If $\theta = \frac{1}{2\sqrt{3}}$, then $\lambda_1^+ = -\pi^2 + O(h^4)$.*

Proof. From Theorem 5.1 we have

$$\lambda_1^+ = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \tag{5.24}$$

where a, b and c and given in (5.5) and where $d = \tan^2(h\pi/2)$. Expanding the right side of (5.24) in a Taylor series with respect to h, we obtain,

$$\lambda_1^+ = -\pi^2 - \frac{1}{24}(12\theta^2 - 1)\pi^4 h^2 - \frac{1}{2880}(720\theta^4 - 200\theta^2 + 13)\pi^6 h^4 + O(h^6) \tag{5.25}$$

so that $\lambda_1^+ = -\pi^2 + O(h^2)$.

Setting $12\theta^2 - 1 = 0$, we obtain $\theta = \frac{\pm 1}{2\sqrt{3}}$ which are the Gauss points in (0,1). Substituting $\theta = \frac{1}{2\sqrt{3}}$ into (5.25), we obtain the desired result,

$$\lambda_1^+ = -\pi^2 - \frac{\pi^6 h^4}{2160} + O(h^6) \Box$$

We now return to the question of the convergence of the TPGADI method when applied to the Discrete Model Problem.

THEOREM 5.3. *For a given set of positive acceleration parameters ρ_k, $k = 1,2,\ldots$, the TPGADI method* (3.2) *applied to the Discrete Model Problem is convergent.*

Proof. The $2M$ and $2N$ generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ and $A_y\mathbf{c} = \lambda B_y\mathbf{c}$ are computed from Theorem 5.1 with $\theta = \frac{1}{2\sqrt{3}}$. A simple calculation shows that the generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ are given by

$$\lambda_0 = \frac{36}{h_x^2} ,$$

$$\lambda_M = \frac{12}{h_x^2} ,$$

$$\lambda_l^{\pm} = \frac{7d + 9 \mp 6\sqrt{d^2 + 90d + 81}}{h_x^2(4d + 3)} , \quad l = 1,\ldots,M-1,$$

where

$$d = \tan^2\left(\frac{l}{M}\,\frac{\pi}{2}\right).$$

Now, since $d > 0$ for all $l = 1,\ldots,M-1$, and since

$$\tan\left(\frac{l}{M}\,\frac{\pi}{2}\right) < \tan\left(\frac{l+1}{M}\,\frac{\pi}{2}\right), \quad l = 1,\ldots,M-2,$$

it follows that the $2M-2$ generalized eigenvalues $\lambda_l^{\pm}$ are distinct, real and positive. Hence, the $2M$ generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ are distinct, real and positive. A similar argument holds for the $2N$ generalized eigenvalues of $A_y\mathbf{c} = \lambda B_y\mathbf{c}$. Convergence now follows immediately from Theorem 3.1 □

For reasonable choices of the basis functions and the collocation points, the generalized eigenvalues of $A_x\mathbf{c} = \lambda B_x\mathbf{c}$ and $A_y\mathbf{c} = \lambda B_y\mathbf{c}$ are accurate approximations to the continuous eigenvalues of L_x and L_y, respectively. In fact, the simple eigenvalues of an m^{th} order ordinary differential equation are approximated within $O(|\Delta|^{2k})$ by collocation at Gauss points with piecewise polynomials of degree $< m+k$ on a set of knots $\Delta = \{0 = t_0 < t_1 < \ldots < t_l = 1\}$, where $|\Delta|$ is the mesh size $|\Delta| = \max_i \Delta t_i$ [de Boor and Swartz, 1980, 1981]. For a large class of operators, the eigenvalues of L_x and L_y are distinct, real and positive, or at least have positive real parts. Hence, we expect the TPGADI method to converge for a large class of elliptic problems for a large class of collocation methods. We apply the TPGADI method to more general discrete elliptic problems in the next section.

6. Computer Implementation and Performance Evaluation

We now consider the performance of a specific computer implementation of the TPGADI method applied to the Hermite bicubic collocation equations. The acceleration parameters ρ_k are computed from the formulas in (5.4); subsequent timings of the TPGADI method include these computations. The acceleration parameters are used in increasing order [Lynch and Rice, 1968]. Although EISPACK [Smith, et. al., 1976] contains routines to solve the generalized eigenvalue problem arising from more general operators, we do not use them. However, we believe that this approach would be cost effective for two and three dimensional problems. The initial iterate, $C^{(0)}$, is always taken to be zero.

The computational complexity can be estimated directly from the analysis of Section 3. The work per x or y direction sweep is estimated from Table 3.1 to be $O(82MN)$ operations. Since the TPGADI method can be a direct method (depending on the choice of the acceleration parameters) in $\min(2M, 2N)$ iterations, it follows that the total work to solve $(A_x \otimes B_y + B_x \otimes A_y)C = F$ is less than or equal to $O(164MN \min(2M, 2N))$ operations. A typical requirement is that $M = N$ so that the total work is $O(328N^3)$.

The matrix $(A_x \otimes B_y + B_x \otimes A_y)$ has dimension $4MN \times 4MN$ and approximate bandwidth $4N$. The work required to factor it using band Gauss elimination with partial pivoting is $O(128MN^3)$ operations which simplifies to $O(128N^4)$ if $M = N$. The bandwidth of

$(A_x \otimes B_y + B_x \otimes A_y)$ can be reduced from $4N$ to $2N$ by using a *finite element ordering* of the collocation points (equations) and basis functions (unknowns) [Dyksen and Rice, 1894a]. Even so, the computer work required to factor the matrix is $O(32MN^3)$ which simplifies to $O(32N^4)$ if $M = N$. Hence, the TPGADI method is asymptotically faster than the straight forward approach of applying band Gauss elimination to $(A_x \otimes B_y + B_x \otimes A_y)$. We experimentally explore the performance of all three solution methods in Example 6.1.

Our implementation of the TPGADI method requires $O(12MN)$ words of computer memory which is nearly optimal since it is only three times the number of unknowns. By contrast, $O(48MN^2)$ words are required to store $(A_x \otimes B_y + B_x \otimes A_y)$ in order to factor it using Gauss elimination with partial pivoting. If the finite element ordering is used, then $O(24MN^2)$ words are required. If $M = N$, then the three methods require $O(12N^2)$, $O(48N^3)$ and $O(24N^3)$ words, respectively.

Before considering any numerical examples, we define two error measures. We denote the Hermite bicubic approximation to the solution u of the elliptic problem at the k^{th} iteration by

$$U^{(k)}(x,y) = \sum_{m=1}^{2M} \sum_{n=1}^{2N} c_{nm}^{(k)} \phi_m(x) \psi_n(y).$$

Two natural error measures are

$$e_C(k) = \frac{\max\limits_{\substack{n=1,\ldots,2N \\ m=1,\ldots,2M}} \left| c_{nm}^{(k)} - c_{nm} \right|}{\max\limits_{\substack{n=1,\ldots,2N \\ m=1,\ldots,2M}} \left| c_{nm} \right|} \quad \text{and} \quad e_U(k) = \frac{\max\limits_{\substack{m=0,\ldots,M \\ n=0,\ldots,N}} \left| U^{(k)}(x_m, y_n) - u(x_m, y_n) \right|}{\max\limits_{\substack{m=0,\ldots,M \\ n=0,\ldots,N}} \left| u(x_m, y_n) \right|} .$$

Note that $e_C(k)$ is the maximum relative error at the k^{th} step in approximating the solution C of the discrete problem, whereas $e_U(k)$ is the maximum error on the grid points at the k^{th} step in approximating the solution u of the continuous problem.

The following numerical results were computed on a VAX 11/780 (UNIX[†], 4.1BSD) with a floating-point accelerator using the Fortran compiler f77 with optimizer in single precision.

EXAMPLE 6.1. Performance of the TPGADI Method with M and N Varied

We solve the Model Problem (5.1) in which f is chosen so that $u(x,y)=x(x-1)(x+2)$ $y(1-y)(3-y)$. The results are summarized in Table 6.1.

Table 6.1
Hermite bicubic collocation and the TPGADI method applied to the Model Dirichlet Problem for $u(x,y)=x(x-1)(x+2)$ $y(1-y)(3-y)$

$N=1/h$	Number of Unknowns	Solution Time (Secs)	$e_C(2N)$	$e_U(2N)$
4	64	0.58	2.4628e-06	4.5776e-07
8	256	4.43	5.3843e-06	8.3923e-07
12	576	14.70	1.3131e-05	1.8142e-06
20	1600	67.57	5.2688e-05	4.4144e-06
28	3136	186.57	1.3512e-04	3.1467e-06

A logarithmic fit of this timing data gives Time $\approx 0.00937N^{2.97}$, which agrees with the theoretical work estimate of $O(328N^3)$ operations.

The error measures in Table 6.1 indicate that the TPGADI method is numerically stable. Since the discretization uses bicubic polynomials, it follows that $e_C(2N)$ and $e_U(2N)$ should be zero within machine round-off. A logarithmic fit of these error measures gives

$$e_C(2N)=9.60*10^{-8}*N^{2.10} \quad \text{and} \quad e_U(2N)=5.53*10^{-8}*N^{1.42}.$$

Thus, not only is machine round-off achieved, but the round-off errors do not grow significantly as N increases.

[†] UNIX is a Trademark of Bell Laboratories

Contrary to intuition, $e_U < e_C$; that is, the error in approximating u is less than that in approximating the coefficients of the basis functions of U. Almost three-fourths of the unknowns correspond to values of U_x, U_y and U_{xy} at the grid points. However, the basis functions associated with them are zero at the grid points so that the error in approximating these unknowns does not contribute to e_U. We include the error measure e_U since it is common and the quantity of interest in many applications.

We now compare the TPGADI scheme as a direct method to band Gauss elimination with partial pivoting by solving the same Model Problem within the ELLPACK[†] system [Rice and Boisvert, 1985]. We obtain a discrete problem using the Hermite bicubic collocation discretization module INTERIOR COLLOCATION which generates the expanded tensor product matrix $(A_x \otimes B_y + B_x \otimes A_y)$ only with a finite element (FE) ordering of the equations and unknowns. We solve the discrete problem by using the band Gauss elimination solution module LINPACK BAND [Dongarra, et. al., 1979]. Moreover, we use a so-called *indexing* module to reorder the linear system produced by INTERIOR COLLOCATION so as to give the tensor product (TP) ordering of the equations and unknowns; that is, to give the exact expanded tensor product linear system $(A_x \otimes B_y + B_x \otimes A_y)C = F$. We solve this form of the discrete problem using LINPACK BAND also. The solution timing results are summarized in Table 6.2.

[†] ELLPACK is a very high level computer language developed at Purdue University for solving second order linear elliptic partial differential equations.

Table 6.2
Solution time (seconds) for LINPACK BAND and the TPGADI method applied to the Discrete Model Problem arising from Hermite bicubic collocation

$N = 1/h$	Number of Unknowns	LINPACK BAND		TPGADI
		FE Ordering	TP Ordering	
4	64	0.40	0.41	0.58
8	256	3.88	6.42	4.43
12	576	15.50	25.82	14.70
20	1600	96.22	164.63	67.57
28	3136	339.23	605.93	186.57

A logarithmic fit of this timing data shows that $\text{Time}_{FE/LB} \approx 0.00307N^{3.46}$, $\text{Time}_{TP/LB} \approx 0.00313N^{3.64}$ and $\text{Time}_{TPGADI} \approx 0.00937N^{2.97}$. We see that even as a **direct method** the TPGADI method is faster than band Gauss elimination. We believe that band Gauss elimination is currently considered to be the best method for solving the collocation equations [Dyksen, et. al., 1984c].

The TPGADI solution time can be reduced significantly by taking less than $2N$ iterations. Most of the accuracy is achieved during the initial iterations by using the ρ_k in increasing order, thereby annihilating the low-frequency components of the error. We solve the same Model Problem with $N = 28$ (3136 unknowns) using varied numbers of iterations of the TPGADI method. The results are given in Table 6.3.

Table 6.3
Hermite bicubic collocation and the TPGADI method applied to the Model Dirichlet Problem with $N = 28$ for $u(x,y)=x(x-1)(x+2)\,y(1-y)(3-y)$

Number of Iterations, K	Solution Time (Secs)	$e_C(K)$	$e_U(K)$
7	21.72	7.8773e-04	1.4985e-06
14	43.76	7.5900e-05	1.4235e-06
21	67.38	6.8314e-05	7.6956e-07
28	89.99	7.2540e-05	7.6956e-07
35	109.74	8.1844e-05	1.4985e-06
42	135.69	8.1539e-05	2.0978e-06
49	163.03	1.2288e-04	2.6223e-06
56	186.57	1.3512e-04	3.1467e-06

Reasonable accuracy is attained with as few as 7 iterations. For this case, the solution time is 21.72 seconds as compared to 339.23 and 605.93 seconds for LINPACK BAND with the two different orderings of the equations and unknowns.

EXAMPLE 6.2. Performance of the TPGADI Method with Varied Partial Differential Operators

We prove in Section 5 that the TPGADI method converges if applied to the Discrete Model Problem. We now solve discrete problems arising from more general separable elliptic operators. We consider varied operators L_x and L_y in (4.1) with f chosen so that $u(x,y)=x(x-1)(x+2)\,y(1-y)(3-y)$. The acceleration parameters are taken to be the Hermite cubic collocation approximations to the eigenvalues of $-u_{xx}$. We use $1/h = M = N = 20$ which gives 1600 unknowns to compute. The results are summarized in Table 6.4.

Table 6.4
The TPGADI method applied to the discrete problem arising from Hermite bicubic collocation with varied partial differential operators L for $1/h = M = N = 20$

$Lu = L_x u + L_y u$	$e_C(40)$	$e_U(40)$
$-u_{xx} - u_{yy}$	5.2688e-05	4.4144e-06
$-u_{xx} - u_{yy} + u_y + u$	5.6071e-05	3.9655e-06
$-u_{xx} - u_{yy} + \sin(y)u_y + e^y u$	4.7869e-05	4.4892e-06
$-u_{xx} - \sin(y)u_{yy} + \cos(y)u_y + u$	7.4526e-05	3.9655e-06
$-u_{xx} - u_{yy} + 1000u$	4.7525e-05	1.3468e-06

Since the discretization is theoretically exact for this choice of u, it follows that $e_C(40)$ and $e_U(40)$ should be zero within machine round-off. The data indicate that the discrete generalized eigenvalues corresponding to L_y cause no ill effects on the iteration process.

EXAMPLE 6.3. Hermite Bicubic Collocation and the TPGADI Method applied to a Problem from Stratospheric Physics

We solve **Problem 6** of [Rice, et. al. 1981b] which is defined by

$$-u_{xx} - u_{yy} + (100 + \cos(2\pi x) + \sin(3\pi y))u = f \quad \text{in } \Omega = [0,1] \times [0,1]$$
$$u = 0 \quad \text{on } \partial\Omega ,$$

where f is chosen so that

$$u(x,y) = -0.31(5.4 - \cos(4\pi x))\sin(\pi x)(y^2 - y)(5.4 - \cos(4\pi y))\left(\frac{1}{1 + p(x,y)^4} - \frac{1}{2}\right)$$

where $p(x,y) = 4(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2$.

For computational purposes we factor the partial differential operator into the sum of

$$L_x u = -u_{xx} + \cos(2\pi x)u$$

and

$$L_y u = -u_{yy} + (100 + \sin(3\pi y))u.$$

The acceleration parameters are computed from (5.4) with $\theta = \frac{1}{2\sqrt{3}}$. Although these acceleration parameters are not the generalized eigenvalues of $A_x c = \lambda B_x c$, we still expect them to work well since $L_x u \approx -u_{xx}$. The results are given in Table 6.5.

Table 6.5
Hermite bicubic collocation and the TPGADI method applied to Problem 6

$N = 1/h$	Number of Unknowns	Solution Time (Secs)	$e_C(N/2)$	$e_U(N/2)$
4	64	0.17	4.9074e-01	4.4000e-02
8	256	1.15	2.6185e-02	3.3113e-03
12	576	3.82	1.5850e-03	7.6008e-04
20	1600	17.48	3.1199e-04	9.8924e-05
28	3136	47.42	1.2299e-04	2.6276e-05

A logarithmic fit of these error measures gives $e_C(N/2) \approx 184N^{-4.39} = 184h^{4.39}$ and $e_U(N/2) \approx 9.08N^{-3.81} = 9.08h^{3.81}$ which agrees with the theoretical convergence rate of $O(h^4)$.

For additional comparison, we also solve Problem 6 within the ELLPACK system using the so-called *triple*[†] module MARCHING ALGORITHM which is a "fast" method designed for separable, self-adjoint elliptic problems [Bank, 1978]. MARCHING ALGORITHM uses a symmetric 5-point finite difference discretization to generate a discrete problem which is then solved using the generalized marching algorithm. The results are summarized in Table 6.6.

[†] In ELLPACK, elliptic problems are typically solved in three phases, *discretization*, *indexing* and *solution*. A *triple* module incorporates all three phases into one module.

Table 6.6
MARCHING ALGORITHM used to solve Problem 6

$1/h$	Number of Unknowns	Solution Time (Secs)	Maximum Error
4	9	0.03	3.1019e-01
8	49	0.05	6.5795e-02
16	225	0.18	1.5340e-02
32	961	0.98	4.0000e-03
64	3969	4.35	1.0000e-03
128	16129	20.22	3.4175e-04
256	65025	85.85	3.2596e-03

Figure 6.1 shows a plot of data from Table 6.5 and Table 6.6. We conclude that in order to achieve a given accuracy for Problem 6, our implementation of Hermite bicubic collocation and the TPGADI method is superior to the implementation of the marching algorithm in the module MARCHING ALGORITHM.

7. Conclusions

We have derived a new Tensor Product Generalized Alternating Direction Implicit (TPGADI) iterative method to solve discrete elliptic problems of the form $(A_1 \otimes B_2 + B_1 \otimes A_2)C = F$. We have demonstrated that a specific implementation of the TPGADI method for the Hermite bicubic collocation equations is fast and numerically stable.

Although we believe that we have implemented our numerical methods well, there exist many obvious routes towards improvement. Since the TPGADI method converges rapidly, we should incorporate some type of automatic stopping criteria into the software. Initial guesses other than zero should be automatically provided; in fact, smooth initial guesses should be provided to accelerate convergence. For Hermite bicubic collocation, the boundary conditions can be interpolated. More general operators can be treated by computing the acceleration

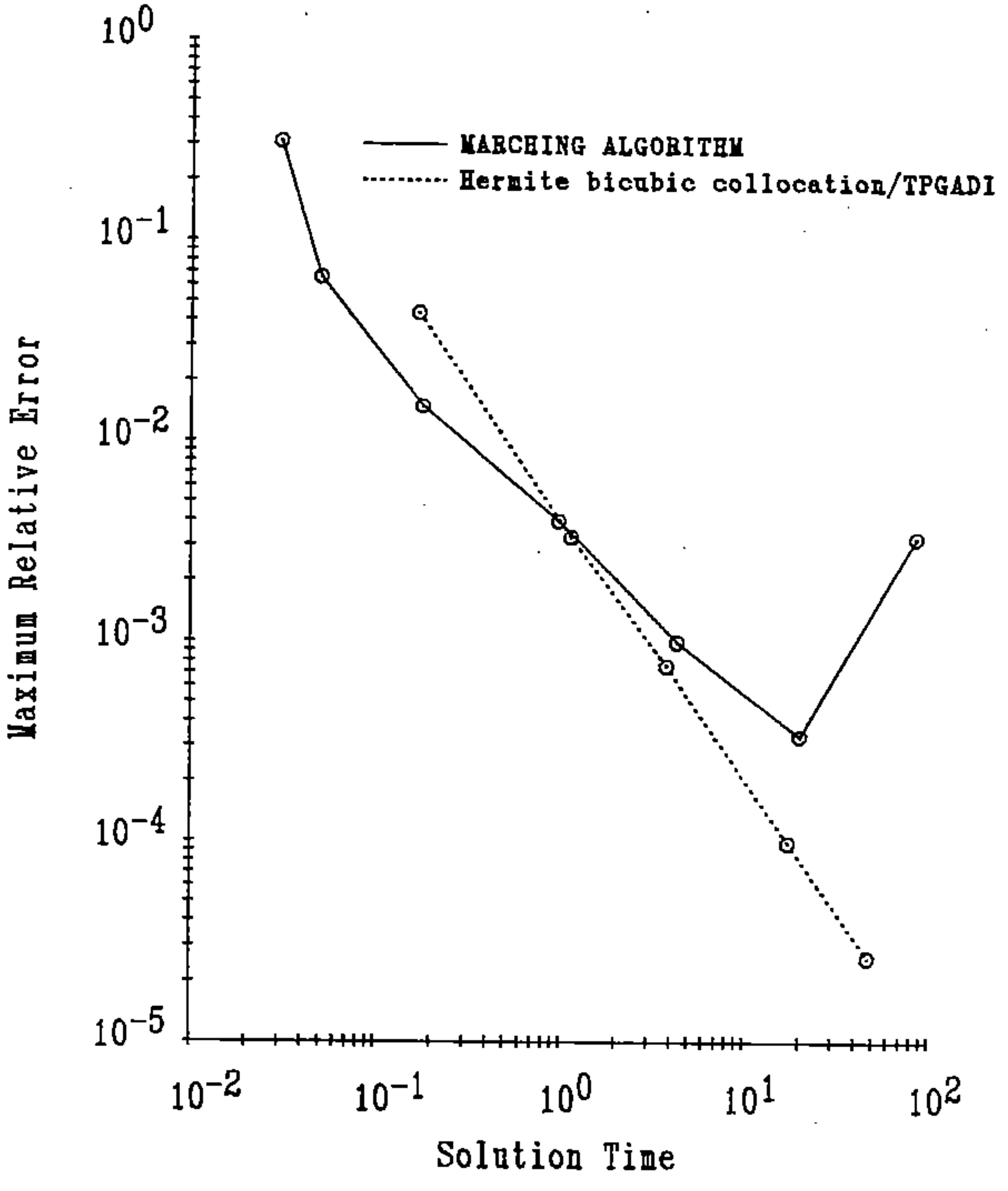

Figure 6.1 Solution Time versus Maximum Relative Error for Hermite bicubic collocation/TPGADI and the MARCHING ALGORITHM applied to Problem 6

parameters by solving the corresponding generalized eigenvalue problem using EISPACK.

The TPGADI method can be used to solve discrete elliptic problems arising from many different discretizations. We have used it in conjunction with the Method of Lines [Dyksen, 1982]. We have implemented within the ELLPACK system the TPGADI method for a discretization method which we call Method of Planes to solve separable elliptic problems with uncoupled boundary conditions on three dimensional rectangular domains [Dyksen, 1984d]. Moreover, we have used the TPGADI method to solve the partial difference equations arising from Dirichlet problems on three dimensional cylindrical domains with holes [Dyksen, 1984c].

References

[1] R. E Bank, "Algorithm 527: A Fortran implementation of the generalized marching algorithm", *ACM Trans. Math. Software*, 4(1978), pp. 165-176.

[2] G. Birkhoff, R. S. Varga and D. M. Young, "Alternating direction implicit methods", *Advances in Computers*, Academic Press, New York, 3(1962), pp. 189-273.

[3] G. Birkhoff and R. E. Lynch, *Numerical Solution of Elliptic Problems*, SIAM, Philadelphia, 1985.

[4] C. de Boor, "Efficient computer manipulation of tensor products", *ACM Trans. Math. Software*, 5(1979), pp. 173-182.

[5] C. de Boor and B. Swartz, "Collocation approximation to eigenvalues of an ordinary differential equation: The principle of the thing", *Math. Comp.*, 35(1980), pp. 679-694.

[6] C. de Boor and B. Swartz, "Collocation approximation to eigenvalues of an ordinary differential equation: Numerical illustrations", *Math. Comp.*, 36(1981), pp. 1-19.

[7] J. J. Dongarra C. B. Moler, J. R. Bunch and G. W. Stewart, *Linpack Users' Guide*, SIAM, Philadelphia, 1979.

[8] J. Douglas, Jr. and H. H. Rachford, Jr., "On the numerical solution of heat conduction problems in two or three space variables", *Trans. Amer. Math. Soc.* 82(1956), pp. 421-439.

[9] W. R. Dyksen, "Tensor Product Generalized Alternating Direction Implicit Methods For Solving Separable Second Order Linear Elliptic Partial Differential Equations", Ph.D. Thesis, Purdue University, 1982.

[10] W. R. Dyksen and J. R. Rice, "A new ordering scheme for the Hermite bicubic collocation equations", in *Elliptic Problem Solvers III*, (G. Birkhoff and A. Schoenstadt, eds.), Academic Press, 1984, pp. 467-480.

[11] W. R. Dyksen and J. R. Rice, "Scale factors for the Hermite bicubic collocation equations", to appear, *Advances in Computer Methods for Partial Differential Equations-V*, (R. Vichnevetsky and R. Stepleman, eds.), IMACS, New Brunswick, NJ, 1984.

[12] W. R. Dyksen, R. E. Lynch, J. R. Rice and E. N. Houstis, "The performance of the collocation and Galerkin methods with Hermite bi-cubics", *SIAM J. Numer. Anal.*, 21(1984), pp. 695-715.

[13] W. R. Dyksen "A tensor product generalized ADI method for the method of planes", Purdue University, Computer Science Department Report CSD-TR 494, October 1984.

[14] W. R. Dyksen "A tensor product generalized ADI method for elliptic problems on cylindrical domains", Purdue University, Computer Science Department Report CSD-TR 495, October 1984.

[15] P. R. Halmos, *Finite-Dimensional Vector Spaces*, 2nd ed., D. Van Nostrand Company, Inc., Princeton, 1958.

[16] E. N. Houstis, "Collocation methods for linear elliptic problems", *BIT*, 18(1978), pp. 301-310.

[17] E. N. Houstis, W. F. Mitchell and J. R. Rice, "Algorithms INTCOL and HERMCOL: Collocation on rectangular domains with bicubic Hermite polynomials', Purdue University, Computer Science Department Report CSD-TR 445, June 1983.

[18] E. N. Houstis, W. F. Mitchell and J. R. Rice, "Collocation software for second order elliptic partial differential equations", Purdue University, Computer Science Department Report CSD-TR 446, June 1983.

[19] R. E. Lynch, J. R. Rice and D. H. Thomas, "Direct solution of partial difference equations by tensor product methods", *Numer. Math.*, 6(1964), pp. 185-199.

[20] R. E. Lynch, J. R. Rice and D. H. Thomas, "Tensor product analysis of partial difference equations", *Bull. Amer. Math. Soc.*, 70(1964), pp. 378-384.

[21] R. E. Lynch, J. R. Rice and D. H. Thomas, "Tensor product analysis of alternating direction implicit methods", *J. Soc. Indust. Appl. Math.*, 13(1965), pp. 995-1006.

[22] R. E. Lynch, and J. R. Rice, "Convergence rates of ADI methods with smooth initial error", *Math. Comp.*, 22(1968), pp. 311-355.

[23] D. W. Peaceman and H. H. Rachford Jr., "The numerical solution of parabolic and elliptic differential equations", *J. Soc. Indust. Appl. Math.*, 3(1955), pp. 28-41.

[24] P. Percell and M. F. Wheeler, "A C^1 finite collocation method for elliptic equations", *SIAM J. Numer. Anal.*, 17(1980), pp. 605-622.

[25] J. R. Rice, *Matrix Computations and Mathematical Software*, McGraw-Hill Book Company, New York, 1981.

[26] J. R. Rice, E. N. Houstis and W. R. Dyksen, "A population of linear, second order, elliptic partial differential equations on rectangular domains, Parts 1 and 2", *Math. Comp.*, 36(1981), pp. 475-484.

[27] J. R. Rice and R. F. Boisvert, *Solving Elliptic Problems using ELLPACK*, Springer-Verlag, New York, 1985.

[28] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema and C. B. Moler, *Matrix Eigensystem Routines - EISPACK Guide*, Springer-Verlag, New York, 1976.

[29] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962.