ITERATIVE SOLUTION OF LARGE SCALE LINEAR SYSTEMS

A thesis submitted to
Lakehead University
in partial fulfillment of the requirements
for the degree of
Master of Science

by

Maurice W. Benson

1973

ACKNOWLEDGMENTS

I wish to thank my supervisor, Professor P. O. Frederickson, for his advice and encouragement during the preparation of this thesis.

I would also like to thank the National Research Council of Canada for support during the preparation of this paper.

ABSTRACT

Several new techniques are given in this thesis for the iterative solution of the linear system $Ax = y$. The class of matrices to which these techniques apply include circulant matrices, band matrices with well behaved inverses, and two dimensional analogues of these. Such matrices arise naturally in spline approximation problems. Our concern is with the iterative process $x^{(m+1)} = (I-BA)x^{(m)} + By$, $m \geq 0$ with B chosen so that $I-BA$ is small (in spectral radius). Thus B is an "approximate inverse" to A and we focus attention on the construction of B.

For the circulant matrix A, starting with Fourier transform theory, we develop several approximate inversion methods, each optimal in its own sense. These procedures include the diagonal block (DBq) method which determines B such that the central $2q+1$ diagonals of $I-BA$ have zero entries, the least-squares (LSq) method which determines the $2q+1$ nonzero row elements of B by a least-squares process in the transform space, and the min-max (MMq) method for symmetric A that produces the B of a particular form such that the spectral radius of $I-BA$ is minimized. Experimental results with test matrices are given with each approximate inversion technique considered.

The DBq and LSq approximate inversion techniques are generalized to handle certain band matrices. The iterative scheme $x^{(m+1)} = (I-BA)x^{(m)} + By$ associated with the approximate inverse B is extended in the manner that the Jacobi iterative method is extended to the successive overrelaxation iterative technique. Experimental results on the test matrices used indicate that some of the methods developed here are capable of outperforming standard techniques by a substantial margin.

Finally, we extend the LSq and DBq techniques to linear operators associated with certain approximation problems on the plane. We develop our notation and approximate inversion techniques for general finite regions on the plane. Experimental work is confined, however, to a two dimensional circulant problem, and results indicate that approximate inversion procedures are well suited to this situation.

CONTENTS

INTRODUCTION

In this thesis, we develop several techniques for approximating the inverse of certain nonsingular $n \times n$ matrices A. These approximate inverses, B, are used to establish iterative processes of the form $x^{(m+1)} = G\,x^{(m)} + k$, $m \geq 0$ to solve the linear system $Ax = y$.

In Chapter 1, we establish our notation and give some fundamental results that serve as a basis for the chapters to follow. The relationship between an approximate inverse and some standard iterative techniques is mentioned. We end this chapter with definitions of computational complexity and effort for our iterative processes. These definitions serve as a basis for comparison of iterative techniques in the chapters to follow.

Chapter 2 deals with approximate inverses for circulant matrices. The circulant situation is recast in terms of convolutions of doubly infinite absolutely summable sequences. This allows us to make use of Fourier transform theory. Based on minimization problems in the transform space, several approximate inversion techniques for circulant matrices are developed. The truncation (TRq) technique determines an approximate inverse B for A according to standard Fourier transform theory. The least-squares (LSq) technique determines B according to a slight modification of the minimization problem associated with

the TRq technique and the diagonal block (DBq) technique determines B according to a modification of the least-squares minimization problem. Finally in Chapter 2 we deal with the min-max (MMq) approximate inversion technique. This technique determines the circulant matrix B of a particular form such that the spectral radius of $G = I-BA$ is minimized.

The LSq and DBq approximate inversion techniques of Chapter 2 are extended in Chapter 3 to certain band matrices whose inverses are well behaved. Such matrices arise naturally in certain approximation problems. Chapter 3 also contains certain extended iterative processes based on approximate inversion techniques. These extensions parallel the extension of the Jacobi iterative technique to the simultaneous overrelaxation, Gauss-Seidel, and successive overrelaxation iterative techniques.

In Chapter 4, we further extend our LSq and DBq approximate inversion techniques to certain linear operators associated with two dimensional approximation problems. A notation is developed that conveniently handles this extension and that lends itself readily to the programming of the algorithms developed. Details are given for two dimensional problems on general finite regions of the plane, but experimental results are restricted to two dimensional circulant interpolation problems on a parallelogram region on the plane.

In Chapter 5 we discuss the concept of an approximate

inverse. We also suggest some further possibilities with two dimensional problems.

CHAPTER 1
FUNDAMENTAL CONCEPTS

1.1. INTRODUCTION

Let X and Y be complex linear spaces and let $A:X \to Y$ be a linear operator. For a given y in the range of A we are interested in solving the linear system $Ax = y$ for $x \in X$. We restrict our attention to finite dimensional X and Y. In the finite case A can be described by a finite matrix and this is sufficient for the discussion of such problems. However, the concept of a linear operator allows more flexibility of notation. This flexibility is especially useful in Chapter 4 where we consider two dimensional problems.

In the next five sections of this chapter we establish our notation and list some standard results which set a background for the work to follow. In Section 1.7 we define our concepts of computational complexity and effort. These concepts provide us with a criterion for comparing iterative processes in the chapters which follow.

1.2. BASIC NOTATION

For the linear space X with basis $\{e^i : i \in I\}$, $x \in X$, and $x = \sum_{i \in I} x_i e^i$ we have, when defined, standard norms such as $\|x\|_\infty = \sup_{i \in I} |x_i|$, $\|x\|_2 = \left(\sum_{i \in I} |x_i|^2 \right)^{1/2}$, and $\|x\|_1 = \sum_{i \in I} |x_i|$. Given the norm $\|\cdot\|_p$ on the linear spaces X and Y, we find

it useful to consider the norm on the linear operator $A:X \to Y$ defined by $\|A\|_p = \sup_{\|x\|_p \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$, where in our case $p = 1,2,\infty$.

Our concern is with linear systems of equations that have unique solutions; that is, with linear operators A which have an inverse A^{-1}. As we are interested in problems that are solvable with the aid of a computer, we concentrate our attention on finite dimensional linear spaces. Therefore, we will be considering linear operators from the n dimensional space X to X. Our problems can be phrased in terms of $n \times n$ matrices and this is the notation we adopt for the greater part of this thesis. However, we keep in mind that this format is just a convenient tool for some of our problems, and in reality what we are dealing with are linear operators. The merit in this attitude becomes apparent in Chapter 4 where we deal with two dimensional problems and where strict devotion to matrix notation, although valid, is awkward and unnatural.

Further to our notation, we denote the spectral radius of the $n \times n$ matrix A by $\rho(A) = \max\{|\lambda| : \lambda$ is an eigenvalue of $A\}$. We use the spectral radius and numbers involving the spectral radius as a basis for most of our comparisons of iterative processes in the chapters which follow.

To lay the foundations necessary for the iterative

methods of the following chapters, we lean heavily on the first few chapters of Varga [15] and Young [16] and much of our notation is adopted from these sources.

1.3. DETERMINATION OF SPECTRAL RADIUS

Much of this section could be stated in the more general notation of linear operators, but as our sole application of these results is in association with $n \times n$ matrices, such a general discussion is not necessary.

For an arbitrary $n \times n$ complex matrix G we have (see Varga [15, p. 65], Young [16, p. 87])

$$\rho(G) = \lim_{m\to\infty} (\| G^m \|_2)^{1/m}. \tag{1.3.1}$$

The norms $\| G\|_\infty$ and $\| G\|_2$ are equivalent and hence

$$\rho(G) = \lim_{m\to\infty} (\| G^m \|_\infty)^{1/m}. \tag{1.3.2}$$

Since $\| G\|_\infty = \max_i \sum_{j=1}^{n} |g_{i,j}|$, for $G = (g_{i,j})$, we have by way of (1.3.2) an easily programed algorithm for finding $\rho(G)$. Appendix B contains an APL program for finding spectral radius by this technique. We use this algorithm extensively when comparing iterative methods. We comment that as the order of the linear system under consideration increases, the execution of this algorithm becomes costly.

1.4. GENERAL ITERATIVE PROCESSES

Our concern is solely with the general iterative method

$$x^{(m+1)} = Gx^{(m)} + k, \quad m \geq 0 \tag{1.4.1}$$

used to solve the linear system

$$Ax = y \tag{1.4.2}$$

where A is an $n \times n$ nonsingular complex matrix, and G is an $n \times n$ iteration matrix. For our purposes we require that (1.4.1) converge for any starting vector $x^{(0)}$ to a vector z independent of $x^{(0)}$ and that this vector z be the unique solution to (1.4.2).

First, (Varga [15, p. 59]), the method (1.4.1) converges to a vector u independent of $x^{(0)}$ if and only if $\rho(G) < 1$. Clearly u satisfies

$$(I-G)u = k. \tag{1.4.3}$$

If (1.4.1) converges to $z = A^{-1}y$ then† $(I-G)A^{-1}y = k$. Conversely if $\rho(G) < 1$ then $I-G$ is nonsingular and $u = (I-G)^{-1}k$. If further $k = (I-G)A^{-1}y$, then $u = A^{-1}y = z$. In summary, for the nonsingular matrix A, (1.4.1) converges to the unique solution $z = A^{-1}y$ of (1.4.2), independent of $x^{(0)}$ if and only if $\rho(G) < 1$ and $k = (I-G)A^{-1}y$.

† This condition on k is developed in Young [16, pp. 65-66] in a slightly more general context than that required for our purposes.

We complete this section by giving a theorem that puts the iterative method (1.4.1) into a slightly modified form which is very appropriate in the context of the chapters which follow.

Theorem 1.4.1. For the nonsingular matrix A, when (1.4.1) converges it converges to the unique solution of $Ax = y$ if and only if there exists a nonsingular matrix B such that $G = I-BA$ and $k = By$.

Proof: (Young [16, p. 68]) When (1.4.1) converges to $z = A^{-1}y$ then $(I-G)z = k$ and $B = (I-G)A^{-1}$ which is nonsingular since $\rho(G) < 1$. Conversely if such a nonsingular B exists and (1.4.1) converges to z then $BAz = By$ and $Az = y$.

The matrix B is acting as an approximate inverse to A. The concept of an approximate inverse is fundamental to all the iterative procedures considered in this thesis.

1.5. CONVERGENCE RATES

If, for a nonsingular $n \times n$ matrix A, we can find a matrix B such that $G = I-BA$, $k = By$ and $\rho(G) < 1$, then (1.4.1) gives us an iterative process which theoretically will provide us with the unique solution to (1.4.2). In practice, however, we may find that the rate of convergence of our process is much too slow to be practical. The convergence rate of the resulting iterative process is one of the major considerations

in determining an acceptable B.

Following Varga [15, p. 62], for the iterative process (1.4.1) used to solve the system (1.4.2) with nonsingular A, we have the error vectors $e^{(m)} = x^{(m)} - z$ where $z = A^{-1}y$. The result that $e^{(m)} = G^m e^{(0)}$, $m \geq 0$ follows immediately and this leads to

$$\| e^{(m)} \|_2 \leq \| G^m \|_2 \, \| e^{(0)} \|_2 .$$

Our interest is in the behaviour of

$$\sigma_m = \left(\frac{\| e^{(m)} \|_2}{\| e^{(0)} \|_2} \right)^{1/m}$$

as $m \to \infty$. We have $\sigma_m \leq \| G^m \|_2^{1/m}$ and (Varga [15, p. 67]) when G is convergent ($\lim_{m\to\infty} G^m$ is the $n \times n$ null matrix) we have

$$\lim_{m\to\infty} - \ell n \, \| G^m \|_2^{1/m} = - \ell n \rho(G) = R_\infty(G) .$$

$R_\infty(G)$ is the asymptotic rate of convergence.

We employ $R_\infty(G)$ with the realization that it is an asymptotic value and may not accurately reflect the initial behaviour of our iterative process. However, it does offer a convenient means of comparing iterative methods and it is to this use that we put it in later chapters.

For our iterative process (1.4.1) (written in terms of Theorem 1.4.1) used to solve (1.4.2), one of our objectives should

be the creation of a matrix B such that $k = By$ and $\rho(G) = \rho(I-BA)$ is as small as possible. In practice, however, one must consider the labour involved in reducing $\rho(G)$ and decide if the energy expenditure required is justified.

1.6. STANDARD METHODS

We give four examples of standard methods which are of the form (1.4.1). Some of these methods are later used as a basis for comparison with the methods developed in subsequent chapters. Following Varga [15, pp. 87-88], we cast these methods in the format of Theorem 1.4.1.

Our concern is with the iterative solution of the system $Ax = y$ where A is an $n \times n$ nonsingular matrix. For nonsingular M, the expression $A = M-N$ represents a splitting of the matrix A and this leads to the iterative process

$$x^{(m+1)} = M^{-1}N\, x^{(m)} + M^{-1}y, \quad m \geq 0.$$

Since this can be written as $x^{(m+1)} = (I-M^{-1}A)x^{(m)} + M^{-1}y$, $m \geq 0$, we see that M^{-1} corresponds to B in Theorem 1.4.1.

We let $A = D-E-F$ where D is a diagonal matrix, and E and F are strictly lower and upper triangular matrices respectively. First we have the Jacobi* method, where we require that D be nonsingular, and we write

* Strictly speaking this is the point Jacobi method as opposed to the block Jacobi method, but as all methods considered in this thesis are point iterative methods, we suppress the word point.

$$x^{(m+1)} = D^{-1}(E+F)x^{(m)} + D^{-1}y, \quad m \geq 0.$$

In this case $M = D$ and $N = E+F$ for our splitting of the matrix A. For brevity we list our examples of standard methods in Table 1.6.1. We include in brackets after the name of each method its abbreviation. These abbreviations provide a convenient notation in later chapters. This is especially true in Chapter 3 where these standard methods are extended.

TABLE 1.6.1
STANDARD ITERATIVE METHODS

METHOD	ITERATION MATRIX G	VECTOR k	MATRIX M	MATRIX N
(1) Jacobi (J)	$D^{-1}(E+F)$	$D^{-1}y$	D	$E+F$
(2) simultaneous overrelaxation (JOR)	$\omega D^{-1}(E+F)+(1-\omega)I$	$\omega D^{-1}y$	$\omega^{-1}D$	$(\omega^{-1}-1)D+E+F$
(3) Gauss-Seidel (GS)	$(D-E)^{-1}F$	$(D-E)^{-1}y$	$D-E$	F
(4) successive overrelaxation (SOR)	$(D-\omega E)^{-1}((1-\omega)D+\omega F)$	$\omega(D-\omega E)^{-1}y$	$\omega^{-1}D-E$	$(\omega^{-1}-1)D+F$

1.7. COMPUTATIONAL COMPLEXITY

Methods developed in subsequent chapters allow us to reduce the spectral radius of the iteration matrix, but often at the expense of increasing the work involved in each iteration. We thus incorporate a measure of this work into our comparison of various techniques. With the understanding that a computer

spends much more time on multiplication than on addition and subtraction, we give

Definition 1.7.1. The computational complexity of the iterative method (1.4.1) is the number of multiplications required to perform a single iteration divided by the order of the system under consideration.

We symbolize our computational complexity by C. When referring to a particular iterative process (for example the GS iterative technique) we denote the associated computational complexity by $C(GS)$. We keep in mind that the computational complexity depends to a great extent on the matrix A of the linear system $Ax = y$ under consideration.

Our interest is in the complexity per iteration and we ignore in our complexity measures the calculations required to establish the iterative process. This provides a convenient measure for comparing iterative processes and when the system $Ax = y$ must be solved with many different values of y, the set-up work decreases in importance. Of course when a problem is being solved on a once only basis it is prudent when choosing a method to include the set-up time among the factors governing a decision.

One method of comparing iterative processes of the form (1.4.1) is to investigate $R_\infty(G)$ for each process, but this

does not give any indication of the computational complexity involved. To include both these measures we define the effort of our techniques.

Definition 1.7.2. The effort, $E(G)$, of the iterative process (1.4.1) is given by

$$E(G) = \frac{C}{R_\infty(G)}.$$

As the effort represents a more complete measure (than just $R_\infty(G)$) of the value of an iterative process in a test situation, we use efforts for comparison purposes in this thesis.

CHAPTER 2
THE CIRCULANT PROBLEM

2.1. NOTATION AND FUNDAMENTAL RESULTS

The iterative methods of Section 1.6 used to solve the linear system $Ax = y$ are given in terms of splittings $A = M-N$ of the matrix A because this formulation leads naturally to the generalizations which follow. For the iterative scheme (1.4.1) used to solve $Ax = y$, we have $G = I-M^{-1}A$ and $k = M^{-1}y$ and for convergence we require $\rho(G) < 1$. Our goal is to make $\rho(G)$ as small as is practically possible. Ultimately if $M = A$ (A nonsingular) then $M^{-1}A = I$ and $\rho(G) = 0$. It is of course undesirable to make $M = A$ since, as noted in Young [16, p. 75], in forming k we are back with the original problem. We may thus think of $B = M^{-1}$ as an approximate inverse to A and each iterative process in Section 1.6 is related to an approximate inversion technique applied to A. It is the concept of an approximate inverse to which we now turn. We begin by considering circulant matrices. In particular we are interested in circulant matrices characterized by the following definition.

Definition 2.1.1. The $n \times n$ band-circulant matrix $A = (a_{i,j})$ of band width $2p + 1$ $(n \geq 2p+1)$ and with band elements $(a_{-p}, \ldots, a_0, \ldots, a_p)$ has

$$a_{i,j} = \begin{cases} a_k, & \text{if } j-i = k \pmod n,\ k \in \{-p,\ldots,p\} \\ 0, & \text{otherwise} \end{cases} \tag{2.1.1}$$

for $1 \le i \le n$, $1 \le j \le n$.

We restate the concept of linear operators represented by circulant matrices in terms of doubly infinite sequences and convolutions in order to take advantage of certain established results. Let M represent the set of all complex $n \times n$ circulant matrices, let ℓ_1 represent the set of all doubly infinite absolutely summable complex valued sequences, and let S_n represent the set of all doubly infinite complex valued periodic sequences of period n. Define $\phi:\ell_1 \to M$ by

$$(\phi(a))_{i,j} = \sum_{k=-\infty}^{\infty} a_{i-j+kn} \tag{2.1.2}$$

where $1 \le i \le n$, $1 \le j \le n$ and where $a \in \ell_1$ is the doubly infinite sequence $\{a_k\}$. Define $\alpha:\mathbb{C}^n \to S_n$ by $\alpha(X)_j = x_{n-j}$ for $0 \le j \le n-1$, where $X = (x_1,\ldots,x_n) \in \mathbb{C}^n$. As this defines the periodic sequence $\alpha(X)$ over one period, by periodic extension, $\alpha(X)_j$ is defined for all integers j. We comment that there is a reversal incorporated into α.

The function ϕ is a homomorphism from the commutative ring $(\ell_1,+,*)$, where $+$ denotes addition of sequences and $*$ denotes convolution of sequences (that is for $x,y \in \ell_1$, $(x*y)_j = \sum_{k=-\infty}^{\infty} x_k y_{j-k}$), onto the commutative ring $(M,+,\cdot)$ where

$+$ denotes matrix addition and $\cdot$ denotes matrix multiplication. It follows that

$$\psi:(\ell_1/\ker\ \phi,+,*) \to (M,+,.) \tag{2.1.3}$$

defined for $a + \ker\ \phi \in \ell_1/\ker\ \phi$ by $\psi(a+\ker\ \phi) = \phi(a)$ is a ring isomorphism. The function α is a linear bijection, and it follows that the vector spaces $(\mathbb{C}^n,+)$ and $(S_n,+)$ are isomorphic. Hence for all $x \in \mathbb{C}^n$, $A \in M$,

$$\alpha(Ax) = \psi^{-1}(A)(\alpha(x)) \tag{2.1.4}$$

where for $a + \ker\ \phi \in \ell_1/\ker\ \phi$ and $x \in S_n$,

$$(a+\ker\ \phi)\ (x) = a * x. \tag{2.1.5}$$

The elements of $\ell_1/\ker\ \phi$ are the equivalence classes under the equivalence relation ρ on ℓ_1 defined for $a,b \in \ell_1$ by $a\rho b$ if and only if $a*x = b*x$ for all $x \in S_n$.

In particular, with the $n \times n$ band-circulant matrix A with band elements $(a_{-p},\ldots,a_p)$ we associate the sequence a given by

$$\ldots,\ 0,\ a_{-p},\ \ldots,\ a_0,\ \ldots,\ a_p,\ 0,\ \ldots \tag{2.1.6}$$

and we let a represent the equivalence class $\psi^{-1}(A)$. Thus if $x,y \in \mathbb{C}^n$, the statement $Ax = y$ is equivalent to $a * \alpha(x) = \alpha(y)$.

With the matrix A of Definition 2.1.1 we associate the expression

$$A(z) = \sum_{j=-\infty}^{\infty} a_j z^j \qquad (2.1.7)$$

where z is a complex variable and the a_j's are the elements of the doubly infinite sequence a associated with A. We see that factoring the polynomial $z^p A(z)$ corresponds to factoring the matrix A into a product of band-circulant matrices. When $z = e^{2\pi it}$, $t \in [0,1]$, (2.1.7) represents the Fourier transform of the sequence a and we write $A(e^{2\pi it}) = \hat{a}(t)$. We comment that since we are considering a to represent a linear operator on the space of all doubly infinite sequences of period n, we might consider the finite Fourier transform with $t \in \left\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}\right\}$. However, as we are interested in large linear systems, we avoid this specialization to a particular finite value of n and consider the continuous Fourier transform with $t \in [0,1]$.

2.2. THE TRUNCATION (TR_q) AND MULTIPLE TRUNCATION (MTR_q) TECHNIQUES

If $\hat{a}(t)$ of the previous section is nonzero for all $t \in [0,1]$ then $\frac{1}{\hat{a}(t)}$ has a Fourier expansion which is absolutely convergent. Let d be the doubly infinite sequence $\{d_k\}$ composed of the coefficients in the Fourier series expansion of $\frac{1}{\hat{a}(t)}$. The d_k's can be found by resolving $\frac{z^p}{z^p A(z)}$ into partial fractions and expanding the resulting terms into series valid

for the unit circle in the complex plane. (We could also find d_k using $d_k = \int_0^1 \frac{e^{-2\pi ikt}}{\hat{a}(t)}\,dt$.) We have $1 = \hat{a}(t)\cdot\hat{d}(t) = (a*d)\hat{}(t)$ and $A^{-1} = \phi(d)$, where ϕ was defined in (2.1.2). We may also use elements from d to create band-circulant approximate inverses to A. Let $B = TR_q(A)$ be the $n \times n$ band-circulant approximate inverse to A of band width $2q + 1$ and with band elements $(b_{-q}, \ldots, b_0, \ldots, b_q)$ where $b_k = d_k$ for $|k| \le q$. We call this the truncation technique. Associated with this approximate inversion process, we have the iterative process $x^{(m+1)} = (I-BA)\,x^{(m)} + By$ used to solve the linear system $Ax = y$.

When the sequence a is $\ldots,0, \frac{1}{4}, 1, \frac{1}{4}, 0, \ldots$ (as is the case with the matrix T_4 of Appendix C), we have $b_k = \frac{2(\sqrt{3} - 2)^{|k|}}{\sqrt{3}}$ which gives $b_0 = 1.16$, $b_1 = -0.309$, $b_2 = 0.0829$, ... and when the sequence a arises from the matrix T_2 of Appendix C, we have $b_0 = 2.21$, $b_1 = -1.37$, $b_2 = 0.759$, $b_3 = -0.409$, $b_4 = 0.219$, $b_5 = -0.117$, $b_6 = 0.0629$, ... In Table 2.2.1 we give some experimental results with the TR_q method applied to T_2 and T_4. Our computational complexity for the iterative process associated with the TR_q method is $C(TR_q) = 2(p+q)+1$.

Table 2.2.1

Results with the TR_q method applied to the test matrices T_2 and T_4 of Appendix C. G is the iteration matrix for the iterative method associated with the approximate inversion method TR_q.

$2q+1$	$TR_q(T_2)$		$TR_q(T_4)$	
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
3	2.22	diverges	0.196	3.1
5	1.20	diverges	0.0526	2.4
7	0.643	29	0.0141	2.1
9	0.344	14	0.00377	2.0
11	0.184	10	0.00101	1.9
13	0.0987	8.2	0.000271	1.8

Closely associated with the TR_q method is a procedure involving the factors of $z^pA(z)$. Knowledge of these factors allows us to write $A = A_1A_2\ldots A_k$ where each A_i, $1 \le i \le k$ is an $n \times n$ band-circulant matrix. We define the multiple truncation approximate inversion technique by $MTR_q(A) = TR_{q_1}(A_1)$ $TR_{q_2}(A_2)\ldots TR_{q_k}(A_k)$ where $q = (q_1,q_2,\ldots,q_k)$. This has the advantage that the values of the q_i's can be varied to fit the requirements of the A_i's. The computational complexity of the iterative process associated with the MTR_q approximate inversion technique is $C(MTR_q) = 2(p+q) + 1$, where $q = q_1+q_2+\ldots+q_k$. Table 2.2.2 contains experimental results with this technique for the matrix T_2 of Appendix C. Both the TR_q and MTR_q

methods are useful; however better methods, and in two cases methods with a good potential for easy generalization to certain non-circulant situations, are developed in the remainder of this chapter.

Table 2.2.2

Results with the MTR_q method applied to T_2 of Appendix C. T_2 is factored into $A_1A_2A_3$ where A_1 has band elements (0.412, 0.990, 0.412); A_2 has band elements (0.120, 0.990, 0.120); and A_3 has band elements (0.00906, 0.990, 0.00906). $MTR_q(T_2) = TR_{q_1}(A_1)TR_{q_2}(A_2)TR_{q_3}(A_3)$. G is the iteration matrix of the associated iterative process and $q = q_1+q_2+q_3$.

$2q+1$	$2q_1+1$	$2q_2+1$	$2q_3+1$	$\rho(G)$	$E(G)$
7	3	3	3	1.23	diverges
9	5	3	3	0.603	30
11	5	5	3	0.667	42
11	7	3	3	0.376	17
13	7	5	3	0.351	18
13	9	3	3	0.187	11
13	5	5	5	0.667	47

2.3. THE LEAST-SQUARES (LS_q) TECHNIQUE

Let A be an $n \times n$ band-circulant matrix of band width $2p+1$ with band elements $(a_{-p},\ldots,a_0,\ldots,a_p)$ and let B be an $n \times n$ band-circulant matrix of band width $2q+1$ with band elements $(b_{-q},\ldots,b_0,\ldots,b_q)$. We have the sequences a (given by $\ldots,0,a_{-p},\ldots,a_p,0,\ldots$) and b (given by $\ldots,0,b_{-q},\ldots,b_q,0,\ldots$)

associated with A and B respectively and we have the Fourier transforms $\hat{a}(t)$ and $\hat{b}(t)$ defined on $[0,1]$. When $\hat{a}(t)$ is nonzero on $[0,1]$, the TR_q method of the previous section determines b, such that

$$\int_0^1 \left| \frac{1}{\hat{a}(t)} - \hat{b}(t) \right|^2 dt \qquad (2.3.1)$$

is minimized.

This leads us to consider the problem of minimizing

$$Q = \int_0^1 |1-\hat{a}(t)\hat{b}(t)|^2 dt = \int_0^1 \left| \frac{1}{\hat{a}(t)} - \hat{b}(t) \right|^2 |\hat{a}(t)|^2 dt \ . \qquad (2.3.2)$$

We are requiring that $\hat{a}(t)\hat{b}(t)$ be the least-squares approximation to $\hat{f}(t) = 1$ on $[0,1]$ (where f is the identity sequence $\ldots,0,1,0,\ldots$) in the hope that this will produce a more optimal approximate inversion technique than the TR_q method.

For convenience we define the reversal operator R on the space of doubly infinite sequences X by $(R(x))_i = x_{-i}$ for $x \in X$. Use is made of the fact that for doubly infinite sequences u and v with $\hat{u}$ and $u*v$ defined we have $\hat{u}^* = \widehat{R(u^*)}$ and $R(u*v) = R(u) * R(v)$ where the superscript $*$ denotes complex conjugate.

The above notation and results are applied to the problem of minimizing Q in equation (2.3.2). We have

$$Q = \int_0^1 (1-\hat{a}(t)\hat{b}(t))(1-\hat{a}^*(t)\hat{b}^*(t))dt, \tag{2.3.3}$$

where $Q = Q(b_{-q},\ldots,b_0,\ldots,b_q)$. We require $\frac{\partial Q}{\partial b_r} = 0$ for $-q \le r \le q$. Now (for $a_i, b_i \in \mathbb{R}$)†

$$\frac{\partial Q}{\partial b_r} = \int_0^1 [(1-\hat{a}\hat{b})(-\hat{a}^* e^{-2\pi irt}) + (1-\hat{a}^*\hat{b}^*)(-\hat{a}\, e^{2\pi irt})]dt = 0$$

which gives

$$2a_{-r} = \int_0^1 (\hat{a}\hat{b}\hat{a}^*\, e^{-2\pi irt} + \hat{a}^*\hat{b}^*\hat{a}\, e^{2\pi irt})dt$$

$$= \int_0^1 (R(a)*a*b)\hat{}\; e^{-2\pi irt}dt + \int_0^1 (R(a)*a*R(b))\hat{}\; e^{2\pi irt}dt$$

$$= (c*b)_r + (c*R(b))_{-r} \tag{2.3.4}$$

where $c = R(a)*a$.

But we have $R(c*R(b)) = R(c)*b$ and $R(c) = c$. Therefore,

$$2a_{-r} = (c*b)_r + (R(c*R(b)))_r$$

$$= 2(c*b)_r,$$

$$\text{and} \quad a_{-r} = (c*b)_r. \tag{2.3.5}$$

The problem has been reduced to a linear system of $2q+1$ equations in $2q+1$ unknowns which in matrix notation reads

† We restrict ourselves to real problems for the remainder of this chapter.

$$\begin{bmatrix} c_0 & c_{-1} & \cdots & c_{-2q} \\ c_1 & c_0 & \cdots & c_{-2q+1} \\ \vdots & & & \vdots \\ c_{2q} & c_{2q-1} & \cdots & c_0 \end{bmatrix} \begin{bmatrix} b_{-q} \\ \vdots \\ b_0 \\ \vdots \\ b_q \end{bmatrix} = \begin{bmatrix} a_q \\ \vdots \\ a_0 \\ \vdots \\ a_{-q} \end{bmatrix}. \quad (2.3.6)$$

Since $R(c) = c$, the above matrix, which we denote by C, is symmetric.

The matrix C can be found directly from $R(a)*a$, but we calculate it in a slightly different manner which again finds application in the more general non-circulant situation of Chapter 3. Let the matrix M be given by

$$M = \begin{bmatrix} a_{-p} & & & \cdots & a_0 & & \cdots & a_p & 0 & & \cdots & 0 \\ 0 & a_{-p} & & \cdots & a_0 & & \cdots & a_p & 0 & & \cdots & 0 \\ 0 & 0 & a_{-p} & \cdots & a_0 & & \cdots & a_p & 0 & \cdots & 0 & \\ \vdots & & & & & & & & & & & \\ 0 & \cdots & 0 & a_{-p} & \cdots & a_0 & \cdots & a_p & 0 & & & \\ 0 & \cdots & 0 & a_{-p} & \cdots & a_0 & \cdots & a_p & & & & \end{bmatrix}. \quad (2.3.7)$$

Here M is a $(2q+1) \times (2(q+p)+1)$ matrix and C satisfies

$$C = MM^T. \quad (2.3.8)$$

A linear system similar to that of (2.3.6) will again occur in Chapter 3, but it will then enjoy a more general interpretation. Equation (2.3.6) provides us with an $n \times n$ band-

circulant approximate inverse B of band width $2q+1$ for an $n \times n$ band-circulant matrix A of band width $2p+1$. We denote this least-squares approximate inversion process and its associated iterative process by LS_q and we write $B = LS_q(A)$. The computational complexity for this iterative process is

$$C(LS_q) = 2(p+q) + 1.$$

Experimental results with this method applied to the matrices T_2 and T_4 of Appendix C are given in Table 2.3.1.

Table 2.3.1

Results with the approximate inversion technique LS_q applied to the circulant test matrices T_2 and T_4 of Appendix C. G is the iteration matrix of the associated iterative process.

$2q+1$	$LS_q(T_2)$		$LS_q(T_4)$	
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
3	0.731	29	0.178	2.9
5	0.489	15	0.0487	2.3
7	0.290	11	0.0131	2.1
9	0.162	8.2	0.00350	2.0
11	0.0879	7.0	0.000939	1.9
13	0.0473	6.2	0.000251	1.8

2.4. THE DIAGONAL BLOCK (DB_q) TECHNIQUE

The least-squares minimization of the previous section suggests that we explore further such minimization problems in

search of practical techniques for getting a band-circulant approximate inverse for certain band-circulant matrices.

We start by considering the problem of minimizing

$$\int_0^1 \left| \frac{1}{\hat{a}(t)} - \hat{b}(t) \right|^2 |\hat{a}(t)| dt \qquad (2.4.1)$$

where $\hat{a}(t)$ and $\hat{b}(t)$ are the same as in (2.3.1).

To simplify the problem we will consider the symmetric case. Consequently, $R(a) = a$ and $R(b) = b$ and $\hat{a}(t)$ and $\hat{b}(t)$ are real valued functions on $[0,1]$. This simplification eliminates expressions involving $(\hat{a}(t)\hat{a}^*(t))^{\frac{1}{2}}$. We further assume, as before, that $\hat{a}(t) \neq 0$ for $t \in [0,1]$. However, since $\hat{a}(t)$ is a real valued function on $[0,1]$, we have $\hat{a}(t) > 0$ on $[0,1]$ or $\hat{a}(t) < 0$ on $[0,1]$ and $|\hat{a}(t)|$ is either $\hat{a}(t)$ or $-\hat{a}(t)$. We seek to minimize

$$Q = \int_0^1 K \left(\frac{1}{\hat{a}(t)} - \hat{b}(t) \right)^2 \hat{a}(t) dt \qquad (2.4.2)$$

where

$$K = \begin{cases} 1 & \text{if } \hat{a}(t) > 0 \text{ on } [0,1] \\ -1 & \text{if } \hat{a}(t) < 0 \text{ on } [0,1] \end{cases}$$

and

$$\hat{a}(t) = a_0 + 2 \sum_{j=1}^{p} a_j \cos 2\pi jt, \qquad (2.4.3)$$

$$\hat{b}(t) = b_0 + 2 \sum_{k=1}^{q} b_k \cos 2\pi kt. \tag{2.4.4}$$

Q is a function of $b_0,\ldots,b_q$, and setting $\frac{\partial Q}{\partial b_r} = 0$ for $0 \le r \le q$ gives, for $r \neq 0$,

$$\frac{\partial Q}{\partial b_r} = \int_0^1 2K\, \hat{a}(t) \left(\frac{1}{\hat{a}(t)} - \hat{b}(t) \right) (-2 \cos(2\pi rt))dt = 0,$$

and this reduces to

$$\int_0^1 \cos(2\pi rt)dt = \int_0^1 \hat{a}(t)\ \hat{b}(t)\ \cos(2\pi rt)dt. \tag{2.4.5}$$

But $r \neq 0$ and we get

$$\int_0^1 (a*b)\hat{}\,(t)\ \cos(2\pi rt)dt = 0 \tag{2.4.6}$$

which reduces to

$$(a*b)_r = 0 \quad \text{for} \quad r \neq 0. \tag{2.4.7}$$

If $r = 0$ then

$$\frac{\partial Q}{\partial b_0} = \int_0^1 -2K\, \hat{a}(t) \left(\frac{1}{\hat{a}(t)} - \hat{b}(t) \right) dt = 0$$

and

$$(a*b)_0 = 1. \tag{2.4.8}$$

We may state our linear system for $0 \le r \le q$ as

$$(a*b)_r = \delta_{r,0} \tag{2.4.9}$$

where

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} .$$

The linear system (2.4.9) represents $q+1$ equations in $q+1$ unknowns. Since $R(a*b) = a*b$, we see that b is being determined such that $(a*b)_k = \delta_{k,0}$ for $|k| \leq q$.

The above interpretation suggests that we extend our process to include non-symmetric cases by requiring that (2.4.9) hold for $-q \leq r \leq q$. This gives us the linear system

$$\begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-2q} \\ a_1 & a_0 & \cdots & a_{-2q+1} \\ \vdots & & & \vdots \\ a_{2q} & a_{2q-1} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} b_{-q} \\ \vdots \\ b_0 \\ \vdots \\ b_q \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \qquad (2.4.10)$$

Given an $n \times n$ band-circulant matrix A of band width $2p+1$ with band elements $(a_{-p},\ldots,a_p)$, we use the linear system (2.4.10) to obtain an $n \times n$ band-circulant approximate inverse for A of band width $2q+1$ and with band elements $(b_{-q},\ldots,b_q)$. We call this approximate inversion method the diagonal block (DB_q) technique and write $B = DB_q(A)$. This method will be generalized in Chapter 3 where it will prove very effective on our test matrices. As well as the advantage that this method is easy to use we have the advantage that our computational complexity is

substantially reduced from that of the LS_q and TR_q methods because of the additional zeros that the DB_q method creates in the matrix $I-BA$.

When solving the linear system $Ax = y$ where A is an $n \times n$ band-circulant matrix of band width $2p+1$ we have,

$$C(DB_q) = 2p,$$

which is independent of q. In practice, however, an increase in q increases the work involved in finding $DB_q(A)$. In Table 2.4.1 we give experimental results with this method for the test matrices T_2 and T_4 of Appendix C.

Table 2.4.1

Results for the approximate inversion method DB_q applied to test matrices T_2 and T_4 of Appendix C. G is the iteration matrix of the associated iterative process.

$2q+1$	$DB_q(T_2)$		$DB_q(T_4)$	
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
3	0.764	22	0.143	1.0
5	0.444	7.4	0.0385	0.61
7	0.243	4.2	0.0103	0.44
9	0.131	3.0	0.00276	0.34
11	0.0703	2.3	0.000740	0.28
13	0.0376	1.8	0.000198	0.23

2.5. THE MIN-MAX (MM_q) TECHNIQUE

For the $n \times n$ band-circulant matrix A of band width $2p+1$ with band elements $(a_{-p},\ldots,a_0,\ldots,a_p)$ we seek the $n \times n$ band-circulant matrix B of band width $2q+1$ with band elements $(b_{-q},\ldots,b_0,\ldots,b_q)$ such that $\rho(I-BA)$ is minimized. Let the first row of $G = I-BA$ be $(g_0,g_1,\ldots,g_{n-1})$. Then (Varga [15, p. 45, problem 13]) the n eigenvalues of G are given by

$$\lambda_j = g_0+g_1\,\phi_j+\ldots+g_{n-1}\,\phi_j^{n-1}, \quad 0 \le j \le n-1 \qquad (2.5.1)$$

where $\phi_j = \exp(2\pi ij/n)$. As in the previous two sections, we have the doubly infinite sequences a and b associated with A and B respectively. We let $c = a*b$ and (2.1.2) gives

$$g_\ell = \sum_{k=-\infty}^{\infty} f_{\ell+nk} - c_{\ell+nk}, \quad 0 \le \ell \le n-1 \qquad (2.5.2)$$

where f is the identity sequence of section 2.3. Since $\phi_j^n = 1$, we have

$$\begin{aligned} \lambda_j &= \hat{f}(j/n) - \hat{c}(j/n) \\ &= 1 - \hat{b}(j/n)\,\hat{a}(j/n) \end{aligned} \qquad (2.5.3)$$

where $\hat{}$ denotes the Fourier transform of Section 2.1. Therefore

$$\rho(I-BA) = \max\{|1-\hat{b}(j/n)\hat{a}(j/n)| : 0 \le j \le n-1\} \qquad (2.5.4)$$

and our goal is to determine $b_{-q},\ldots,b_q$ such that we minimize this maximum.

As in Section 2.1, it is not our intention to tailor our

results to a specific n and so since we are interested in large linear systems we consider the problem of determining the b_k's to minimize

$$|| 1-\hat{b}(t)\hat{a}(t) ||_\infty . \tag{2.5.5}$$

We assume that $R(a) = a$ which makes $\hat{a}(t)$ and $\hat{b}(t)$ real valued functions, and we use an exchange algorithm to minimize

$$|| 1-\left(a_0 + 2 \sum_{j=1}^{p} a_j \cos 2\pi jt\right)\left(b_0 + 2 \sum_{k=1}^{q} b_k \cos 2\pi kt\right)||_\infty . \tag{2.5.6}$$

As in previous cases we assume $\hat{a}(t) \neq 0$ for $t \in [0,1]$. We have $\hat{c}(t) = \hat{a}(t)\hat{b}(t)$, and when we wish to consider $\hat{c}$ explicitly as a function of $b_0,b_1,\ldots,b_q$, we write $\hat{c}(b_0,\ldots,b_q,t)$. Since for any integer ℓ, $\cos 2\pi\ell(1-t) = \cos 2\pi\ell t$, we have $\hat{c}(t) = \hat{c}(1-t)$ and the min-max approximation of $\hat{c}(t)$ to 1 on $[0,\frac{1}{2}]$ gives us the same $b_0,\ldots,b_q$ that we would get if we used the whole interval $[0,1]$. Our problem now is to determine $b_0,\ldots,b_q$ to minimize

$$|| 1-\hat{c}(t) ||_\infty = \sup_{t\in[0,\frac{1}{2}]} |1-\hat{c}(t)| . \tag{2.5.7}$$

First we prove that an exchange method will give us the unique $\hat{c}(t)$ which satisfies this requirement. To accomplish this we require the following defintion (Meinardus [12, p. 16]).

Definition 2.5.1. Let T be a compact set and let $C(T)$ denote the space of all continuous real or complex valued functions on

T. A linear subspace V of C(T) of finite dimension n is said to fulfill the Haar condition if for every f in V where $f \neq 0$, f vanishes at no more than n-1 points of T.

Now (Meinardus [12, p. 16, pp. 105-111]) if the Haar condition is satisfied for a linear subspace V of the real space C[a,b] then for any $f \in C[a,b]$ there is a unique function $f_V \in V$ such that $v = f_V$ minimizes

$$|| f(t)-v(t) ||_\infty$$

for all possible $v \in V$. Furthermore f_V can be found iteratively by an exchange method (which we will describe shortly). This motivates the following theorem.

Theorem 2.5.1. For the min-max problem of (2.5.7) the Haar condition is satisfied.

Proof. For this problem $T = [0,\frac{1}{2}]$ and our linear subspace V of C(T) has as a basis

$$\{\hat{a}(t) \cos 2\pi kt: 0 \leq k \leq q\}$$

Our space V has dimension q+1 and if $v \in V$ then there exist numbers v_k such that

$$\begin{aligned} v(t) &= \sum_{k=0}^{q} v_k \hat{a}(t) \cos 2\pi kt \\ &= \hat{a}(t) \sum_{k=0}^{q} v_k \cos 2\pi kt. \end{aligned} \tag{2.5.8}$$

Now

$$\cos n\alpha = 2 \cos(n-1)\alpha \cos \alpha - \cos(n-2)\alpha \qquad (2.5.9)$$

and hence there exist numbers P_k such that

$$v(t) = \hat{a}(t) \sum_{k=0}^{q} P_k (\cos 2\pi t)^k \qquad (2.5.10)$$

Since $\hat{a}(t) \neq 0$ for $t \in [0,1]$, the number of zeros of v in $[0,\frac{1}{2}]$ equals the number of zeros of

$$\sum_{k=0}^{q} P_k (\cos 2\pi t)^k$$

in $[0,\frac{1}{2}]$. But this is a polynomial of degree q in $\cos 2\pi t$ and hence there are at most q values of $\cos 2\pi t$ which make this zero. Since we are considering the interval $[0,\frac{1}{2}]$ this implies that there are at most q zeros of v in $[0,\frac{1}{2}]$ and the Haar condition is satisfied.

Since we assumed $R(a) = a$, we are dealing with real valued functions on $[0,\frac{1}{2}]$ and because the Haar condition holds we can use an exchange algorithm to obtain the values of $b_0,\ldots,b_q$.

In employing an exchange method we approximate the continuous min-max fit on $[0,\frac{1}{2}]$ required by the above discussion by a min-max fit on a set of equally spaced points in $[0,\frac{1}{2}]$. This makes the programing of the exchange method easier and gives an accurate enough answer for our purposes. (In our numerical

experiments we use 101 points including 0 and $\frac{1}{2}$.)

Let the interval $[0,\frac{1}{2}]$ be divided into N equal subintervals and let the points of division be $0 = t_0, t_1,\ldots,t_N = \frac{1}{2}$ where if $i < j$ then $t_i < t_j$. The value N is an initial parameter for our exchange method and the N+1 points thus generated remain fixed throughout the application of the algorithm. The first step of the exchange method used to solve for $b_0,\ldots,b_q$ is to pick q+2 points from $\{t_0,\ldots,t_N\}$. (This requires that $N \geq q+1$.) We choose these points (starting with $t_0 = 0$) as equally spaced as possible in the interval $[0,\frac{1}{2}]$, although the exchange algorithm would converge for any initial set of q+2 points from $\{t_0,\ldots,t_N\}$. We denote these q+2 points by $t_{i_0},\ldots,t_{i_{q+1}}$ and we determine $b_0^1,\ldots,b_q^1$ such that $\hat{c}_1(t) = \hat{c}(b_0^1,\ldots,b_q^1,t)$ approximates 1 on $\{t_{i_0},\ldots,t_{i_{q+1}}\}$ with an error of constant magnitude and alternating sign on these q+2 points. This is accomplished by solving the linear system

$$1 - \hat{c}_1(t_{i_u}) + (-1)^u h = 0, \quad 0 \leq u \leq q+1 \qquad (2.5.11)$$

for $b_0^1,\ldots,b_q^1$, h where $|h|$ is the magnitude of the error at $t_{i_0},\ldots,t_{i_{q+1}}$. Let

$$a_{u,v} = (\cos 2\pi v t_{i_u})\left(a_0 + 2\sum_{j=1}^{n} a_j \cos 2\pi j t_{i_u}\right) \qquad (2.5.12)$$

for $0 \leq v \leq q$, $0 \leq u \leq q+1$. In matrix notation, the linear system (2.5.11) reads

$$\begin{bmatrix} a_{0,0} & 2a_{0,1} & \cdots & 2a_{0,q} & 1 \\ a_{1,0} & 2a_{1,1} & \cdots & 2a_{1,q} & -1 \\ \vdots & & & \vdots & \\ a_{q+1,0} & 2a_{q+1,1} & \cdots & 2a_{q+1,q} & (-1)^{q+1} \end{bmatrix} \begin{bmatrix} b_0^1 \\ b_1^1 \\ \vdots \\ b_q^1 \\ h \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} . \quad (2.5.13)$$

After solving this system of equations, we calculate $\hat{c}_1(t_0),\ldots,\hat{c}_1(t_N)$ and locate a point in $\{t_0,\ldots,t_N\}$ where the maximum deviation of $\hat{c}_1(t)$ from 1 occurs. Call this point t_p. If $|1 - \hat{c}_1(t_p)| \leq |h|$ then the process is finished and $b_0 = b_0^1,\ldots,b_q = b_q^1$. If $|1 - \hat{c}_1(t_p)| > |h|$ then an exchange is made by the following standard technique. We augment the set $t_{i_0},\ldots,t_{i_{q+1}}$ by t_p and then discard one of the $t_{i_0},\ldots,t_{i_{q+1}}$ such that the deviations of $\hat{c}_1(t)$ from 1 on the remaining $q+2$ points alternate in sign. The above procedure is repeated on this new set of points. This is continued until a min-max fit of $\hat{c}(t)$ to 1 over the initial $N+1$ points is obtained. An APL program for executing this algorithm is given in Appendix A.

We denote the min-max approximate inversion technique by MM_q and when we are solving the linear system $Ax = y$ where A is an $n \times n$ band-circulant matrix of band width $2p+1$, the computational complexity of our associated iterative process is

$$C(MM_q) = 2(p+q) + 1.$$

In Table 2.5.1 we give experimental results for the MM_q method with matrices T_2 and T_4 of Appendix C. We comment that $MM_q(T_4) = DB_q(T_4)$ for $q = 1,2,\ldots,6$, however this is not true for T_2. No method can surpass the MM_q procedure when the only criterion is the spectral radius of the iteration matrix for the associated iterative process. However, compared to the DB_q method, the MM_q method creates many more nonzero terms in the iteration matrix. The relatively small computational complexity and the ease of implementation of the DB_q method favour it over the MM_q technique.

Table 2.5.1

Results with the MM_q approximate inversion technique applied to the matrices T_2 and T_4 of Appendix C. G is the iteration matrix of the associated iterative process.

2q+1	$MM_q(T_2)$		$MM_q(T_4)$	
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
3	0.620	19	0.143	2.6
5	0.363	11	0.0384	2.2
7	0.199	8.1	0.0103	2.0
9	0.108	6.7	0.00276	1.9
11	0.0576	6.0	0.000739	1.8
13	0.0309	5.5	0.000198	1.8

We end this section with a theorem giving a bound on $\rho(I-BA)$ for certain symmetric $n \times n$ band-circulant matrices A when B is determined such that the expression in (2.5.5) is minimized.

Theorem 2.5.2. Let A be an $n \times n$ symmetric band-circulant matrix with band elements $(a_{-p},\ldots,a_0,\ldots,a_p)$ such that $\hat{a}(t) \neq 0$ for $t \in [0,1]$. If B is the $n \times n$ symmetric band-circulant matrix with band elements $(b_{-q},\ldots,b_0,\ldots,b_q)$ chosen such that $\| 1 - \hat{b}(t)\hat{a}(t) \|_\infty$ is minimized, then

$$\rho(I-BA) \leq \frac{1}{2\pi q}\left(1 + \frac{\pi^2}{2}\right) \| \frac{d}{dt}\left(\frac{1}{\hat{a}(t)}\right) \|_\infty \| \hat{a}(t) \|_\infty. \tag{2.5.14}$$

Proof. Let β be the doubly infinite sequence $\ldots,0,\beta_{-q},..,\beta_0,\ldots,\beta_q,0,\ldots$ with $\beta_j = \beta_{-j}$ for all integers j and with β_j, $0 \leq j \leq q$ chosen such that $\| \frac{1}{\hat{a}(t)} - \hat{\beta}(t) \|_\infty$ is minimized. Since, by (2.5.4), $\rho(I-BA)$ is the maximum of $|1 - \hat{b}(t)\hat{a}(t)|$ on $\left\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}\right\}$, we have

$$\begin{aligned}
\rho(I-BA) &\leq \| 1 - \hat{b}(t)\hat{a}(t) \|_\infty \\
&\leq \| 1 - \hat{\beta}(t)\hat{a}(t) \|_\infty \\
&\leq \| \hat{a}(t) \|_\infty \| \frac{1}{\hat{a}(t)} - \hat{\beta}(t) \|_\infty \\
&\leq \frac{1}{2\pi q}\left(1 + \frac{\pi^2}{2}\right) \| \frac{d}{dt}\left(\frac{1}{\hat{a}(t)}\right) \|_\infty \| \hat{a}(t) \|_\infty
\end{aligned}$$

where the last line follows from D. Jackson, see Meinardus [12, p. 54].

2.6. SUMMARY OF TECHNIQUES FOR CIRCULANT MATRICES

In Table 2.6.1, we compare the efforts for some of the iterative processes mentioned in this chapter applied to linear

systems involving the matrices T_2 and T_4 of Appendix C. The DB_q technique is clearly the superior method for these test matrices even if one does not take into account its ease of implementation compared to some of the other techniques such as the MM_q procedure.

Table 2.6.1

Comparison of efforts for some iterative processes applied to linear systems involving the matrices T_2 and T_4 of Appendix C.

Iterative method		Effort with matrix indicated	
		T_2	T_4
J		diverges	2.9
GS		26	1.8
SOR		15	2.2
TR_q, $2q+1$ =	3	diverges	3.1
	5	diverges	2.4
	7	29	2.1
	9	14	2.0
LS_q, $2q+1$ =	3	29	2.9
	5	15	2.3
	7	11	2.1
	9	8.2	2.0
DB_q, $2q+1$ =	3	22	1.0
	5	7.4	0.61
	7	4.2	0.44
	9	3.0	0.34
MM_q, $2q+1$ =	3	19	2.6
	5	11	2.2
	7	8.1	2.0
	9	6.7	1.9

CHAPTER 3
APPROXIMATE INVERSES FOR CERTAIN BAND MATRICES

3.1. INTRODUCTION AND NOTATION

Our concern in this chapter is with nonsingular $n \times n$ band matrices whose inverses are well approximated by band matrices. We say that the $n \times n$ matrix $A = (a_{i,j})$, $1 \le i \le n$, $i \le j \le n$, is a band matrix of band width $2p+1$ if $|i-j| > p$ implies $a_{i,j} = 0$, and our objective is to determine an $n \times n$ band matrix $B = (b_{i,j})$ of band width $2q+1$ such that B is in some sense an approximation to A^{-1}. For our purposes, it is essential that $\rho(I-BA) < 1$. It is also desirable that q be small compared to n, and that B be relatively easy to obtain.

For $I = BA = A^T B^T$ we must have

$$M_i^T b_i = f_i, \quad 1 \le i \le n \tag{3.1.1}$$

where b_i is the vector

$$b_i = (b_{i,i-s},\ldots,b_{i,i},\ldots,b_{i,i+t}) \tag{3.1.2}$$

with $s = \min(q,i-1)$ and $t = \min(q,n-i)$;

$$M_i = \begin{bmatrix} a_{i-s,i-s-u} & \cdots & a_{i-s,i+t+v} \\ \vdots & & \vdots \\ a_{i+t,i-s-u} & \cdots & a_{i+t,i+t+v} \end{bmatrix} \tag{3.1.3}$$

with $u = \min(p,i-s-1)$ and $v = \min(p,n-i-t)$ and where

$$f_i = (f_{i,i-k},\ldots,f_{i,i},\ldots,f_{i,i+\ell}) \tag{3.1.4}$$

with $k = \min(p+q,i-1)$, $\ell = \min(p+q,n-i)$ and with $f_{i,j} = 1$ if $i = j$ and 0 otherwise for $1 \le i \le n$, $1 \le j \le n$. In general (3.1.1) represents a set of overdetermined systems of equations which cannot be satisfied exactly. However, as we demonstrate in the next two sections, in certain cases these systems can be approximately satisfied quite successfully.

3.2. GENERALIZED LEAST-SQUARES TECHNIQUE

For the matrix A of Section 3.1 we determine the $n \times n$ band matrix B of band width $2q+1$ such that the Euclidean norm of $G = I-BA$ is minimized. That is for $G = (g_{i,j})$ we minimize†

$$\|G\|_E = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} g_{i,j}^2 \right)^{1/2}. \tag{3.2.1}$$

This is equivalent to minimizing

$$Q_i(b_i) = (M_i^T b_i - f_i)^T (M_i^T b_i - f_i) \tag{3.2.2}$$

independently for $1 \le i \le n$, where the M_i, f_i, and b_i were defined in Section 3.1.

We comment that the above n minimization problems are local in nature in that b_i is determined from entries in the band of A that occur in rows close to the i'th row. Of course

† for the real case

such local techniques are not capable of producing good approximate inverses for all nonsingular band matrices. However, as experimental results in this and the next section indicate, in certain cases such methods work quite well.

To solve the minimization problems of (3.2.2), we take partial derivatives of $Q_i(b_i)$ with respect to the components of b_i and equate these derivatives to zero. This gives

$$M_i M_i^T b_i = M_i f_i, \quad 1 \leq i \leq n. \tag{3.2.3}$$

We observe from equations (2.3.6), (2.3.7) (2.3.8), and (3.2.3) that the LS_q method of Chapter 2 determines the $n \times n$ band-circulant approximate inverse B for the $n \times n$ band-circulant matrix A such that $\| I-BA\|_E$ is minimized and no confusion results if we also denote the approximate inversion procedure of this section by LS_q. The procedure LS_q is now defined for both $n \times n$ band and $n \times n$ band-circulant matrices.

Given the linear system $Ax = y$, where A is an $n \times n$ band matrix of band width $2p+1$, we denote the least-squares approximate inverse of A by $B = LS_q(A)$. The computational complexity for the associated iterative method $x^{(m+1)} = (I-BA)x^{(m)} + By$, $m \geq 0$, is, strictly speaking, $2(p+q) + 1 - \frac{(p+q)(p+q+1)}{n}$. The term involving $1/n$ in this computational complexity decreases in importance as n becomes large. As our interest is in large linear systems, and as computational complexity is at best only

an estimate, we ignore terms in $1/n$ in our complexities. This produces a somewhat high estimate of the computational complexity in our non-circulant test situations when $n = 20$, however, the results obtained are more in line with those expected for larger systems. Thus for the least-squares method we have

$$C(LS_q) = 2(p+q) + 1.$$

In Table 3.2.1 we give experimental results with the LS_q method for the matrices T_1, T_3, and T_5 of Appendix C.

Table 3.2.1

Results with the approximate inversion method LS_q applied to T_1, T_3, and T_5 of Appendix C. G is the iteration matrix of the associated iterative process.

$2q+1$	$LS_q(T_1)$		$LS_q(T_3)$		$LS_q(T_5)$	
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
3	0.995	1800	0.522	7.7	0.650	16
5	0.977	470	0.112	3.2	0.298	7.4
7	0.909	140	0.0223	2.4	0.231	7.5
9	0.741	45	0.00551	2.1	0.118	6.1
11	0.464	22	0.00143	2.0	0.0422	4.7
13	0.206	12	0.000382	1.9	0.0215	4.4

3.3. GENERALIZED DIAGONAL BLOCK TECHNIQUE

The approximate inversion method DB_q of Chapter 2 is generalized in this section to a method for $n \times n$ band matrices.

Let $A = (a_{i,j})$ be an $n \times n$ band matrix of band width $2p+1$. We seek an $n \times n$ band matrix $B = (b_{i,j})$ of band width $2q+1$ such that for $I-BA = G = (g_{i,j})$, $|i-j| \leq q$ implies $g_{i,j} = 0$. We let b_i be defined as in Section 3.1 and let

$$D_i = \begin{bmatrix} a_{i-s,i-s} & \cdots & a_{i-s,i+t} \\ \vdots & & \vdots \\ a_{i+t,i-s} & \cdots & a_{i+t,j+t} \end{bmatrix} \tag{3.3.1}$$

where as in Section 3.1, $s = \min(q,i-1)$ and $t = \min(q,n-i)$. Let

$$d_i = (f_{i,i-s},\ldots,f_{i,i},\ldots,f_{i,i+t}) \tag{3.3.2}$$

where the $f_{i,j}$ are defined as in Section 3.1. We require that

$$D_i^T b_i = d_i, \quad 1 \leq i \leq n. \tag{3.3.3}$$

This defines the generalized diagonal block technique and no confusion results if we symbolize this process by DB_q and write $B = DB_q(A)$. The DB_q method, like the LS_q method, is a local approximate inversion procedure.

When dealing with the linear system $Ax = y$ where A is an $n \times n$ nonsingular band matrix of band width $2p+1$, our computational complexity for the iterative process associated with the above approximate inversion technique is

$$C(DB_q) = 2p.$$

This expression for our computational complexity takes advantage of the central band of zeros in $I-BA$. As with $C(LS_q)$, we ignore terms in $1/n$. In Table 3.3.1, we give experimental results with the DB_q technique applied to the matrices T_1, T_3, and T_5 of Appendix C.

Table 3.3.1

Results with the approximate inversion method DB_q applied to T_1, T_3, and T_5 of Appendix C. G is the iteration matrix of the associated iterative process.

$2q+1$	$DB_q(T_1)$		$DB_q(T_3)$		$DB_q(T_5)$	
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
3	0.914	67	0.277	1.6	0.784	16
5	0.537	9.7	0.0768	0.78	0.229	2.7
7	0.298	5.0	0.0206	0.52	0.206	2.5
9	0.159	3.3	0.00552	0.38	0.0958	1.7
11	0.0953	2.6	0.00148	0.31	0.0370	1.2
13	0.0446	1.9	0.000399	0.26	0.0333	1.2

3.4. A GENERALIZATION OF THE SUCCESSIVE OVERRELAXATION ITERATIVE METHOD

Unlike the Gauss-Seidel and the successive overrelaxation iterative methods, the procedures we have developed so far do not use the available components of $x^{(m+1)}$ when finding $x^{(m+1)}$. Also, we have not made use of relaxation factors yet, and strictly speaking, our methods should only be compared with

the Jacobi method. In this section we extend our methods to procedures which use available components of $x^{(m+1)}$ when finding $x^{(m+1)}$ and to procedures which employ a relaxation factor.

Given the linear system $Ax = y$, where A is an $n \times n$ nonsingular band or band-circulant matrix, and the approximate inversion technique IT (IT is for example the DB_q technique for some q), we write $B = IT(A)$ and have the associated iterative process

$$x^{(m+1)} = (I-BA)x^{(m)} + By, \quad m \geq 0. \tag{3.4.1}$$

Let $H = I-BA = H_L + H_U$ where H_L is an $n \times n$ strictly lower triangular matrix and H_U is an $n \times n$ upper triangular matrix. We start with (3.4.1) instead of the Jacobi method and parallel the development of the simultaneous overrelaxation, Gauss-Seidel, and successive overrelaxation methods from the Jacobi method. For the real number ω, the parallel to the simultaneous overrelaxation method is

$$x^{(m+1)} = \omega(Hx^{(m)} + By) + (1-\omega)x^{(m)}, \quad m \geq 0. \tag{3.4.2}$$

We denote this iterative process by JOR(IT). The parallel to the Gauss-Seidel method is

$$x^{(m+1)} = H_L x^{(m+1)} + H_U x^{(m)} + By, \quad m \geq 0$$

or

$$x^{(m+1)} = (I-H_L)^{-1} H_U x^{(m)} + (I-H_L)^{-1} By, \quad m \geq 0. \qquad (3.4.3)$$

We denote this iterative process by GS(IT). The parallel to the successive overrelaxation process is, for the real number ω,

$$x^{(m+1)} = \omega(H_L x^{(m+1)} + H_U x^{(m)} + By) + (1-\omega)x^{(m)}, \quad m \geq 0.$$

This may be written as

$$x^{(m+1)} = (I-\omega H_L)^{-1}(\omega H_U + (1-\omega)I)x^{(m)} + \omega(I-\omega H_L)^{-1}By, \; m \geq 0. \qquad (3.4.4)$$

We denote this iterative process by SOR(IT). For consistency we denote the iterative process of (3.4.1) by J(IT). As one would expect, for $\omega = 1$ the SOR(IT) process reduces to the GS(IT) process.

Next we consider the special case where the $n \times n$ matrix A has nonzero diagonal elements. Let D be the $n \times n$ matrix which is zero off its diagonal and whose diagonal equals the diagonal of A. It follows that $DB_0(A) = D^{-1}$, and hence the $J(DB_0)$, $JOR(DB_0)$, $GS(DB_0)$, and $SOR(DB_0)$ methods are equivalent to the J, JOR, GS, and SOR methods respectively. (This does not hold in general for the LS_0 technique.)

The SOR(IT) method presents the added problem of determing the optimal relaxation factor ω_b. The problems of the uniqueness of ω_b and local minima for $\rho((I-\omega H_L)^{-1}(\omega H_u+(1-\omega)I))$ as a function of ω which are not absolute minima have not been

investigated.

When A is a band matrix of band width $2p+1$, we have the following computational complexities;

$$C(GS(LS_q)) = 2(p+q) + 1,$$

$$C(SOR(LS_q)) = C(GS(LS_q)) + 1,$$

$$C(GS(DB_q)) = 2p,$$

$$C(SOR(DB_q)) = C(GS(DB_q)) + 1.$$

Again we neglect terms in $1/n$ in our computational complexities. Experimentally, we deal with both the GS(IT) and SOR(IT) methods. The former does not involve the determination of ω_b, however once ω_b is found, the SOR(IT) method is, in certain cases, substantially superior to the GS(IT) method. All our relaxation factors were determined experimentally. Experimental results with the $GS(LS_q)$ and $GS(DB_q)$ methods are given in Table 3.4.1 for the matrices T_1, T_3, and T_5 of Appendix C, and in Table 3.4.2 results with the $SOR(LS_q)$ and $SOR(DB_q)$ methods are given for the same matrices.

Table 3.4.1

Results with the $GS(LS_q)$ and $GS(DB_q)$ iterative techniques for the matrices T_1, T_3, and T_5 of Appendix C. G is the iteration matrix in each case.

Method	2q+1	Results with matrix indicated					
		T_1		T_3		T_5	
		$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$
$GS(LS_q)$,	3	0.995	1800	0.484	6.9	0.530	11
	5	0.976	450	0.0736	2.7	0.138	4.5
	7	0.904	130	0.00580	1.7	0.0979	4.7
$GS(DB_q)$,	3	0.835	33	0.0769	0.78	0.627	8.6
	5	0.280	4.7	0.00589	0.38	0.0520	1.4
	7	0.0890	2.5	0.000425	0.26	0.0424	1.3

Table 3.4.2

Results with the $SOR(LS_q)$ and $SOR(DB_q)$ iterative techniques for the matrices T_1, T_3, and T_5 of Appendix C. G is the iteration matrix in each case and ω_b is the optimal relaxation factor determined experimentally to ± 0.005.

Method	2q+1	Results with matrix indicated								
		T_1			T_3			T_5		
		ω_b	$\rho(G)$	$E(G)$	ω_b	$\rho(G)$	$E(G)$	ω_b	$\rho(G)$	$E(G)$
$SOR(LS_q)$,	3	2.195	0.988	830	1.310	0.306	5.1	1.300	0.277	6.2
	5	2.005	0.948	220	1.035	0.0390	2.5	1.055	0.0579	3.5
	7	1.825	0.815	68	1.005	0.00506	1.9	1.040	0.0431	3.8
$SOR(DB_q)$,	3	1.425	0.463	9.1	1.020	0.0208	0.77	1.230	0.370	5.0
	5	1.085	0.0897	2.9	1.0015	0.00150	0.46	1.005	0.420	1.6
	7	1.025	0.0273	1.9	1.00015	0.000150	0.34	1.010	0.0197	1.3

3.5. HYBRID TECHNIQUES

We observe that in certain cases it is to our advantage to vary from row to row the number of nonzero elements in our approximate inverse. For example, the first and last few rows of the

inverse to T_1 are the most difficult to approximate with a band matrix. This suggests that we use more nonzero elements in the first and last few rows of our approximate inverse than for the remaining rows. To illustrate, we might use $DB_{q_1}(T_1)$ values for the first and last few rows of our approximate inverse and values from $DB_{q_2}(T_1)$, where $q_2 < q_1$ for the remaining rows. We also have the option of using different approximate inversion techniques for various parts of our approximate inverse. We may, for example use the min-max approximate inversion technique on the circulant portion of T_1 and another method for the ends of the band in T_1. We will say that an approximate inversion technique is a hybrid technique if it uses a varying criterion to determine the elements of its approximate inverse.

Hybrid techniques for getting an approximate inverse to a band matrix A provide more flexibility than the procedures mentioned so far. For efficient application of a hybrid technique, knowledge of the more "difficult" portions of A^{-1} must be available. When applying a hybrid technique, we must decide on the techniques to be employed for the various portions of the inverse being created and we must decide on the number of nonzero elements to be allowed in each row of the approximate inverse.

Experimental results indicate that for the matrices T_1 and T_3 of Appendix C we can take the exact inverse of a small (say 8×8) version of these matrices and using data from

this small inverse (and if desired data from one of the band-circulant approximate inversion methods of Chapter 2) we can patch together an approximate inverse to the given matrix. Good experimental results were obtained with such techniques. It is not our intention, however, to pursue in detail here the creation of hybrid approximate inversion techniques.

3.6. SUMMARY OF TECHNIQUES FOR BAND MATRICES

We emphasize that the techniques given in this chapter to produce an $n \times n$ band matrix B that acts as an approximate inverse to a given $n \times n$ band matrix A are intended for situations where A^{-1} is well approximated by zero entries away from a central band. In Table 3.6.1 we compare the efforts for some of the iterative processes mentioned in this chapter applied to linear systems involving the matrices T_1, T_3, and T_5 of Appendix C. For these test matrices, the DB_q method stands out as being the most useful. It is superior to the J, GS, SOR, and LS_q methods and it serves as a better basis for an extended method than does the LS_q procedure.

Table 3.6.1

Comparison of efforts for some iterative processes applied to linear systems involving the matrices T_1, T_3, and T_5 of Appendix C.

Iterative method		Effort with matrix indicated		
		T_1	T_3	T_5
J		diverges	3.1	diverges
GS		57	1.8	12
SOR		13	2.4	11
LS_q, $2q+1$ =	3	1800	7.7	16
	5	470	3.2	7.4
	7	140	2.4	7.5
DB_q, $2q+1$ =	3	67	1.6	16
	5	9.7	0.78	2.7
	7	5.0	0.52	2.5
$GS(LS_q)$, $2q+1$ =	3	1800	6.9	11
	5	450	2.7	4.5
	7	130	1.7	4.7
$GS(DB_q)$, $2q+1$ =	3	33	0.78	8.6
	5	4.7	0.38	1.4
	7	2.5	0.26	1.3
$SOR(LS_q)$, $2q+1$ =	3	830	5.1	6.2
	5	220	2.5	3.5
	7	68	1.9	3.8
$SOR(DB_q)$, $2q+1$ =	3	9.1	0.77	5.0
	5	2.9	0.46	1.6
	7	1.9	0.34	1.3

CHAPTER 4

TWO DIMENSIONAL APPROXIMATION PROBLEMS

4.1. INTRODUCTION

One application of the iterative processes considered so far occurs in the approximation of a function of one variable by a spline $S = \sum_{k=0}^{n} a_k S_k$, where the S_k are translates of the basic piecewise cubic spline S_0 of Appendix C. While this is useful, a more interesting problem is the approximation of functions of two variables. Our domain is now a region in the plane and our approximating function is a linear combination of translates of a two dimensional extension of S_0 such as the one outlined in Appendix D.

We begin our attack on the problem by constructing a mesh over our two dimensional region as in Figure 4.1.1. We require that all the basic regions defined by this mesh be congruent parallelograms. If $\theta = 90°$ then

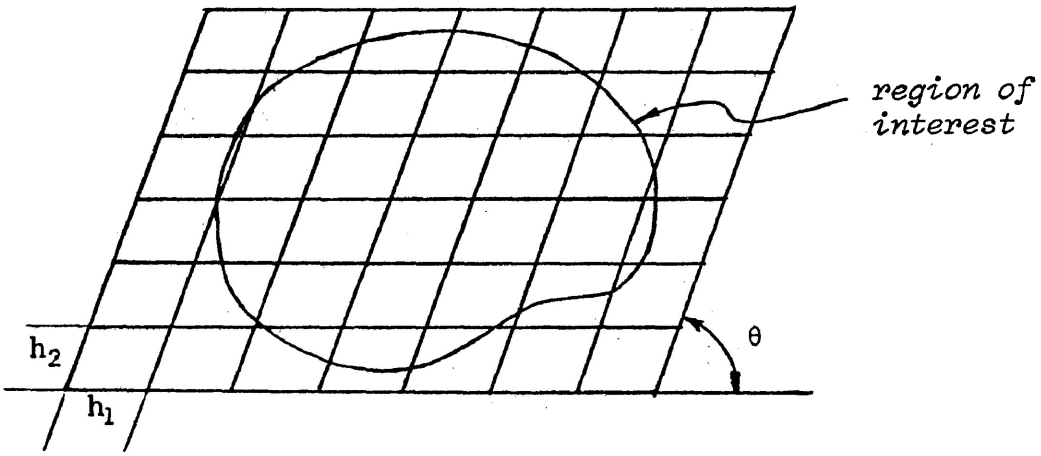

Figure 4.1.1

we have a rectangular mesh and if further $h_1 = h_2$ then we have a square mesh. In our numerical examples $h_1 = h_2$ and $\theta = 60°$. Our mesh points are designated by ordered pairs of integers and we restrict our attention to sets of mesh points whose boundary points lie on a parallelogram in the plane. Thus for an irregular region we may be designating mesh points which are put to no direct use. However, the inclusion of these points allows us to develop a simple and natural notation for the problem. Of course in practice such points are not included in computer programs where such inclusion would result in an undue waste of storage space.

We designate the mesh point in the i'th row of points from the top and the j'th column (inclined at θ degrees to the horizontal) of points from the left by (i,j). We let $S_{i,j}$ denote a basic two dimensional spline centered at the mesh point (i,j). To facilitate the following discussion, we let Ω denote the set of all mesh points (i,j) such that the spline $S_{i,j}$ is being used in the given approximation problem. For example, in the least squares approximation problem, we use $S_{i,j}$ if it has nonzero values in the region in question.

Given the function f defined on a region in the plane, our objective is to determine $x_{i,j}$ for $(i,j) \in \Omega$ such that

$$\sum_{(i,j)\in\Omega} x_{i,j} S_{i,j} \tag{4.1.1}$$

approximates f over the region in question. It is convenient to consider the $x_{i,j}$ as elements of the matrix

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \tag{4.1.2}$$

where if $(i,j) \notin \Omega$ we set $x_{i,j} = 0$. We let

$$X_\Omega = \{X = (x_{i,j}) : X \text{ is an } m \times n \text{ matrix and } x_{i,j} = 0 \text{ if } (i,j) \notin \Omega\}.$$

We observe that X_Ω is a subspace of the linear space of all $m \times n$ matrices.

We assume that our approximation problems have unique solutions. This occurs, for example, in least-squares approximation problems and in interpolation problems if sufficient boundary conditions are present as they are in the periodic problem of Section 4.6. The approximation problem of (4.1.1) may thus be stated in terms of the linear system

$$AX = Y \tag{4.1.3}$$

where $X, Y \in X_\Omega$ and A is a nonsingular linear operator from X_Ω to X_Ω.

We observe that if the splines $S_{i,j}$ have small support in the region under consideration then each $y_{i,j}$ of Y for $(i,j) \in \Omega$ depends through A on only a few nearby elements of X (that is on elements $x_{k,\ell}$, $(k,\ell) \in \Omega$ where the mesh

point (k,ℓ) is geometrically near the mesh point (i,j)). In the next section we develop a convenient notation for linear operators of this form. Linear operators of this type arise in Section 4.6 where we interpolate a doubly periodic function with a linear combination of quartic triangular splines. We observe that if the linear system of (4.1.3) were written in standard matrix notation, we would get a matrix which, although sparse, is neither band nor band-circulant in nature.

4.2. NOTATION AND FUNDAMENTAL CONCEPTS

Because of the limited overlap of elements in the set of translates of our basic two dimensional spline, each equation in (4.1.3) involves only elements in a relatively small portion of A. This observation leads to the following formulation.

Let $A = (A_{i,j})$ be an $m \times n$ array, each of whose elements is a $(2p+1) \times (2p+1)$ matrix $A_{i,j}$ given by

$$A_{i,j} = \begin{bmatrix} a_{i,j}^{-p,-p} & \cdots & a_{i,j}^{-p,p} \\ \vdots & & \\ a_{i,j}^{0,-p} & \cdots\ a_{i,j}^{0,0}\ \cdots & a_{i,j}^{0,p} \\ \vdots & & \\ a_{i,j}^{p,-p} & \cdots & a_{i,j}^{p,p} \end{bmatrix}. \qquad (4.2.1)$$

We could be more general and not require each $A_{i,j}$ to have the same dimension, however, the above notation is sufficient for the purposes of this chapter. The array A, which we use to describe a higher dimensional analog of the band matrices considered in the previous chapter, can be used to define the linear operator of (4.1.3). Since only the $x_{i,j}$'s of X in (4.1.3) with $(i,j) \in \Omega$ have any effect on an approximation problem under consideration, we assume for integers t, u with $|t| \leq p$, $|u| \leq p$, that $a_{i,j}^{t,u} = 0$ if $(i+t, j+u) \notin \Omega$. We also assume that $A_{i,j}$ is the $(2p+1) \times (2p+1)$ null matrix if $(i,j) \notin \Omega$. This last assumption assures that there is a one-to-one correspondence between the elements $y_{i,j}$ of Y for which $(i,j) \in \Omega$ and the equations in the linear system (4.1.3).

For simplicity in stating the following definition, we define $x_{k,\ell} = 0$ if any of the following occur: $k < 1$, $k > m$, $\ell < 1$, $\ell > n$.

Definition 4.2.1. For the $m \times n$ matrix X define AX to be the $m \times n$ matrix given by

$$(AX)_{i,j} = \sum_{r=-p}^{p} \sum_{s=-p}^{p} a_{i,j}^{r,s} x_{i+r,j+s}. \tag{4.2.2}$$

It follows that A is a linear operator from the space X_Ω to the space X_Ω, and the linear system (4.1.3) can be conveniently given in this notation. Indeed, our definition of the

way A operates on an $m \times n$ matrix $X \in X_\Omega$ is a higher dimensional analog of the multiplication of a band matrix and a vector.

4.3. MULTIPLICATION OF THE LINEAR OPERATORS $A = (A_{i,j})$ AND $B = (B_{i,j})$

Let $A = (A_{i,j})$ and $B = (B_{i,j})$ be operators defined on the space X_Ω as in Section 4.2. Let each $A_{i,j}$ be a $(2p+1) \times (2p+1)$ matrix and let each $B_{i,j}$ be a $(2q+1) \times (2q+1)$ matrix. We seek the linear operator $C = (C_{i,j})$ such that for any $m \times n$ matrix $X \in X_\Omega$

$$CX = B(AX). \tag{4.3.1}$$

In terms of the previous section, the operator C can be represented by an $m \times n$ array of $(2(p+q)+1) \times (2(p+q)+1)$ matrices $C_{i,j}$ following the format of (4.2.1) such that for integers t, u with $|t| \leq p+q$, $|u| \leq p+q$, the elements $c_{i,j}^{t,u}$ of $C_{i,j}$ are given by

$$c_{i,j}^{t,u} = \sum_{r=-q}^{q} \sum_{s=-q}^{q} b_{i,j}^{r,s} \, a_{i+r,j+s}^{t-r,u-s} \tag{4.3.2}$$

where if $|t-r| > p$, or $|u-s| > p$ or $(i+r,j+s) \notin \Omega$ then $a_{i+r,j+s}^{t-r,u-s} = 0$. We observe that if $(i,j) \notin \Omega$ then $C_{i,j}$ is the $(2(p+q)+1) \times (2(p+q)+1)$ null matrix and if $(i+t, j+u) \notin \Omega$, then $c_{i,j}^{t,u} = 0$.

4.4. APPROXIMATE INVERSES FOR THE OPERATOR $A = (A_{i,j})$

We find it convenient to define our identity operator on X_Ω by the $m \times n$ array $I(r) = (I_{i,j}(r))$ where each $I_{i,j}(r)$ is a $(2r+1) \times (2r+1)$ matrix in the format of (4.2.1) with elements $f_{i,j}^{t,u}(r)$ given by

$$f_{i,j}^{t,u}(r) = \begin{cases} 1, & \text{if } t = u = 0 \text{ and } (i,j) \in \Omega \\ 0, & \text{otherwise} \end{cases} . \quad (4.4.1)$$

We are concerned with linear operators $A = (A_{i,j})$ whose inverses are well approximated by linear operators $B = (B_{i,j})$ where the dimensions of $A_{i,j}$ and $B_{i,j}$ are small compared to the dimensions of the array A. Specifically for the linear operator A given by Definition 4.2.1, we seek a linear operator B of the form used in Section 4.3 such that $BA = C = (C_{i,j})$ in some sense approximates $I(p+q)$. Ideally $B = A^{-1}$ and

$$C_{i,j} = I_{i,j}(p+q), \quad 1 \le i \le m, \quad 1 \le j \le n. \quad (4.4.2)$$

It is not in general possible to satisfy these overdetermined systems exactly and we must be content with an approximate solution. Of course the elements $c_{i,j}^{t,u}$ of $C_{i,j}$ automatically satisfy (4.4.2) if $(i+t, j+u) \notin \Omega$, and when determining B, the only pertinent equations arising from (4.4.2) are those for which $(i+t, j+u) \in \Omega$.

Two approximate inversion techniques of the previous

chapter generalize nicely to two dimensional problems. First we consider a generalization of the least-squares technique. Let $G = (G_{i,j}) = I(p+q) - BA$. (We define this subtraction by $G_{i,j} = I_{i,j}(p+q) - (BA)_{i,j}$.) We minimize (for the real case)

$$Q(B) = \left(\sum_{i,j,t,u} \left| g_{i,j}^{t,u} \right|^2 \right)^{1/2} \tag{4.4.3}$$

where $g_{i,j}^{t,u}$ is an element of $G_{i,j}$. This is equivalent to minimizing

$$Q_{i,j}(B) = \mathrm{tr}([I_{i,j}(p+q) - (BA)_{i,j}]^T [I_{i,j}(p+q) - (BA)_{i,j}]) \tag{4.4.4}$$

independently for each $(i,j) \in \Omega$, where tr denotes the trace operator. In a similar manner to the least-squares procedure of Chapter 3, the minimization problems of (4.4.4) are local in nature in that $Q_{i,j}(B)$ depends only on $B_{i,j}$ which is determined from data in matrices in $A = (A_{i,j})$ whose subscripts correspond to mesh points in Ω that are geometrically near the mesh point (i,j) in the plane.

The value of $B_{i,j}$ that minimizes (4.4.4) is the least-squares solution to the overdetermined linear system represented by

$$(BA)_{i,j} = I_{i,j}(p+q).$$

After writing this overdetermined linear system in standard matrix

notation (which for reasons of space we do not do here), we see that our problem is handled by the procedures of Section 3.2. No ambiguity arises if we denote this approximate inversion process by LS_q and write $B = LS_q(A)$.

The diagonal block approximate inversion technique also generalizes to the linear operator $A = (A_{i,j})$. For this procedure we require that

$$g_{i,j}^{t,u} = f_{i,j}^{t,u}(p+q) - (BA)_{i,j}^{t,u} = 0 \quad \text{if} \quad |t| \leq q, \ |u| \leq q. \quad (4.4.5)$$

Determination of the $B_{i,j}$ for $(i,j) \in \Omega$ according to the linear systems arising from (4.4.5) gives our approximate inverse B by the diagonal block technique for two dimensional problems. No confusion results if we denote this approximate inversion technique by DB_q and write $B = DB_q(A)$. We observe that this procedure, like our two dimensional extension of the least-squares technique, is a local technique. Another advantage of the DB_q technique is that this procedure supplies a great number of zero entries in the matrices of $G = I - BA$, and thus is capable of reducing the computational complexity from that of the LS_q technique. We observe, however, that in certain cases it is more economical not to form the $G_{i,j}$ but rather to apply A and B individually in the iterative process

$$X^{(m+1)} = (I-BA)X^{(m)} + BY$$

$$= X^m - BAX^{(m)} + BY,\ m \geq 0.$$

This is the case for example when $p = 2$ and $q = 1$, however, if $p = 1$ and $q = 1$ it is more advantageous (in terms of computational complexity) to form G.

4.5. THE TWO DIMENSIONAL CIRCULANT PROBLEM

We modify slightly the work of the previous sections to treat the two dimensional extension of the circulant problem handled in Chapter 2. This sets a background for the experimental work of the following section where we consider a two dimensional periodic interpolation problem. Again we consider linear operators from the space of all $m \times n$ matrices to the space of all $m \times n$ matrices. We represent our two dimensional extension of the band-circulant matrices of Chapter 2 by an $m \times n$ array $A = (A_{i,j})$ of $(2p+1) \times (2p+1)$ matrices $A_{i,j}$ following the format of (4.2.1). In the circulant case all the $A_{i,j}$ are equal. For the $m \times n$ matrix $X = (x_{i,j})$, AX is the $m \times n$ matrix whose elements are defined by

$$(AX)_{i,j} = \sum_{r=-p}^{p} \sum_{s=-p}^{p} a_{i,j}^{r,s} (E(X))_{p+i+r,p+j+s} \qquad (4.5.1)$$

for $1 \leq i \leq m$, $1 \leq j \leq n$, where $E(X)$ is a $(m+2p) \times (n+2p)$ periodic extension of X with $(E(X))_{p+i,p+j} = x_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq n$, and with $(E(X))_{k,\ell} = (E(X))_{t,u}$ if $k \equiv t \pmod{m}$ and

$\ell \equiv u \pmod{n}$.

Next we consider multiplication of our two dimensional circulant operators. Let the $m \times n$ array $B = (B_{i,j})$, where each $B_{i,j}$ is a $(2q+1) \times (2q+1)$ matrix, define a two dimensional circulant operator according to (4.5.1) and let $C = BA$. We observe that C may be represented by an $m \times n$ array of $(2(p+q)+1) \times (2(p+q)+1)$ matrices $C_{i,j}$ with

$$c_{i,j}^{t,u} = \sum_{r=-q}^{q} \sum_{s=-q}^{q} b_{i,j}^{r,s} \; a_{i,j}^{t-r,u-s} \tag{4.5.2}$$

where $|t| \leq p+q$, $|u| \leq p+q$ and $a_{i,j}^{t-r,u-s} = 0$ if $|t-r| > p$ or $|u-s| > p$.

Following the previous section we have the least-squares approximate inversion method (denoted by LS_q) which requires that $B_{i,j}$ be the least-squares solution to the overdetermined system $(BA)_{i,j} = I_{i,j}(p+q)$. We also have the diagonal block approximate inversion method (denoted by DB_q) which requires that

$$(BA)_{i,j}^{t,u} = \begin{cases} 1, & \text{if } t = u = 0 \\ 0, & \text{otherwise} \end{cases}$$

for $|t| \leq q$, $|u| \leq q$.

We list in Appendix F, FORTRAN programs for determining $DB_q(A)$ and $LS_q(A)$ by the successive overrelaxation iterative technique. We comment that our algorithms for finding these approximate inverses make use of the notation developed in this

chapter. Thus we do not require explicit matrix statements of the linear systems whose solutions give our approximate inverses.

In the next section, we make use of the following technique to determine the spectral radius of two dimensional circulant linear operators. Let $G = (G_{t,u})$ be an $m \times n$ array of $(2p+1) \times (2p+1)$ matrices $G_{t,u}$ following the format of (4.2.1), and let G define a circulant linear operator. We observe that the eigenvectors of G are the $m \times n$ matrices $\Phi^{r,s}$, $1 \le r \le m$, $1 \le s \le n$ whose elements $\phi^{r,s}_{t,u}$, $1 \le t \le m$, $1 \le u \le n$ are given by $\phi^{r,s}_{t,u} = \exp(2\pi irt/m)\exp(2\pi isu/n)$. It follows that the eigenvalues of G are given by

$$\lambda_{r,s} = \sum_{k=-p}^{p} \sum_{\ell=-p}^{p} g^{k,\ell}_{t,u} \exp(2\pi irk/m)\exp(2\pi is\ell/n) \qquad (4.5.3)$$

and $\rho(G) = \max\{|\lambda_{r,s}| : 1 \le r \le m, 1 \le s \le n\}$. We consider the symmetric case where $g^{k,\ell}_{t,u} = g^{-k,-\ell}_{t,u}$ and

$$\lambda_{r,s} = g^{0,0}_{t,u} + 2 \sum_{k=1}^{p} g^{k,0}_{t,u} \cos \frac{2\pi rk}{m} + 2 \sum_{k=-p}^{p} \sum_{\ell=1}^{p} g^{k,\ell}_{t,u} \cos 2\pi\left(\frac{rk}{m} + \frac{s\ell}{n}\right). \qquad (4.5.4)$$

A FORTRAN program to determine the spectral radius of a symmetric two dimensional circulant operator is given in Appendix E.

4.6. APPLICATION TO A SPLINE INTERPOLATION PROBLEM

We consider the mesh of Figure 4.1.1 with $\theta = 60°$ and

$h_1 = h_2 = h$ and we consider a function $f(x,y)$ defined on the plane in reference to the coordinate system of Figure 4.6.1. Our goal is to interpolate the function $f(x,y)$ in the parallelogram

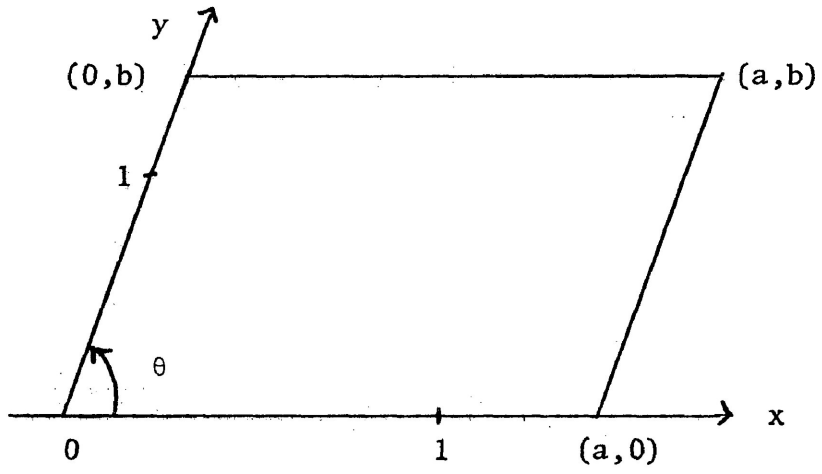

Figure 4.6.1

defined by $(0,0)$, $(0,b)$, (a,b), $(a,0)$ in the special case when for all (x,y),

$$f(x,y) = f(x+a,y) = f(x,y+b). \qquad (4.6.1)$$

We further assume that for positive integers m, n; $a = nh$ and $b = mh$. We then have an $m \times n$ matrix X of variables. The resulting linear system for the interpolation problem using translates of the basic spline of Appendix D is

$$AX = Y$$

where $A = (A_{i,j})$ is the circulant linear operator with

$$A_{i,j} = \frac{1}{12}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 6 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad (4.6.1)$$

for $1 \le i \le m$, $1 \le j \le n$. Y is the $m \times n$ matrix formed by the values of $f(x,y)$ on the mesh points of the $m \times n$ mesh under consideration.

For the approximate inverse B to A, we consider the iterative process

$$X^{(p+1)} = (I-BA)X^{(p)} + BY, \quad p \ge 0. \qquad (4.6.2)$$

For comparison purposes we determine $\rho(I-BA)$ for various approximate inverses, and we test the process (4.6.2) for various values of B on an actual linear system. We make use of

$$\delta^{(p)} = \max\{|x_{i,j}^{(p+1)} - x_{i,j}^{(p)}| : 1 \le i \le m, 1 \le j \le n\}. \qquad (4.6.3)$$

In Appendix G, we give a FORTRAN program for carrying out our two dimensional iterative procedures in the circulant case.

Next we give some examples of approximate inverses for $q = 1$. With A defined by (4.6.1), the $B_{i,j}$ of $LS_1(A)$ are given by

$$B_{i,j} = \begin{bmatrix} -0.245 & -0.287 & 0.0959 \\ -0.287 & 2.25 & -0.287 \\ 0.0959 & -0.287 & -0.245 \end{bmatrix} \qquad (4.6.4)$$

and the elements of $DB_1(A)$ are given by

$$\begin{bmatrix} -0.282 & -0.302 & 0.101 \\ -0.302 & 2.30 & -0.302 \\ 0.101 & -0.302 & -0.282 \end{bmatrix}. \qquad (4.6.5)$$

The hexagonal shape of the basic spline under consideration suggests that we consider least-squares and diagonal block approximate inverses that reflect this geometric property of our basic spline. In particular for $q = 1$, we might consider the least-squares and diagonal block approximate inverses with $b_{i,j}^{-1,1} = b_{i,j}^{1,-1} = 0$. With this additional constraint, the elements of the least-squares approximate inverse are

$$\begin{bmatrix} -0.255 & -0.255 & 0 \\ -0.255 & 2.225 & -0.255 \\ 0 & -0.255 & -0.255 \end{bmatrix}, \qquad (4.6.6)$$

and the elements of the diagonal block approximate inverse are

$$\begin{bmatrix} -0.286 & -0.286 & 0 \\ -0.286 & 2.286 & -0.286 \\ 0 & -0.286 & -0.286 \end{bmatrix}. \qquad (4.6.7)$$

For comparison purposes we also consider the quasi-inverse $B = (B_{i,j})$, $1 \le i \le m$, $1 \le j \le n$ with each $B_{i,j}$ given by

$$\frac{1}{12}\begin{bmatrix} -1 & -1 & 0 \\ -1 & 18 & -1 \\ 0 & -1 & -1 \end{bmatrix}. \qquad (4.6.8)$$

This is an exact inverse for the interpolation of functions of degree three or less (Frederickson [4]).

In Table (4.6.1) we give experimental results with the LS_q and DB_q approximate inversion techniques. In Table (4.6.2) we give experimental results with the seven point approximate inverses of (4.6.6) and (4.6.7), and with the quasi-inverse of (4.6.8). We observe that the quasi-inverse produces excellent results ($p_1 = 2$) for the well behaved Y of Table 4.6.2 in spite of the relatively high spectral radius of the associated I-BA. For comparison purposes, we comment that the iteration operator $G = I-BA$ for the Jacobi iterative process ($B = DB_0(A)$) has spectral radius one.

Table 4.6.1

Results with the LSq and DBq approximate inversion techniques used with (4.6.2) to solve $AX = Y$ where $Y = (y_{i,j})$ is a 25×35 array given by $y_{i,j} = \sin \frac{2\pi i}{25} \sin \frac{2\pi j}{35}$. We start with $X^{(0)} = Y$, and p_1 is the smallest number such that $\delta^{(p_1)} < 10^{-6}$.

2q+1	LSq			DBq		
	$\rho(I-BA)$	Effort	p_1	$\rho(I-BA)$	Effort	p_1
3	0.237	11	6	0.275	11	7
5	0.0649	12	4	0.0821	8.8	4
7	0.0163	14	2	0.0216	6.9	3

Table 4.6.2

Results with the seven point least-squares approximate inverse, the seven point diagonal block approximate inverse and the quasi-inverse used with the iterative technique (4.6.2) to solve $AX = Y$. The definitions of $X^{(0)}$, Y and p_1 are the same as in the previous table.

Approximate inverse	$\rho(I-BA)$	Effort	p_1
seven point least-squares	0.307	12	8
seven point diagonal block	0.429	14	10
quasi-inverse	0.562	24	2

CHAPTER 5
SUMMARY AND CONCLUSIONS

5.1. THE CONCEPT OF AN APPROXIMATE INVERSE

The concept of an approximate inverse and its relationship to the iterative process $x^{(m+1)} = G\,x^{(m)} + k$, $m \geq 0$, is central to the thesis. Conte and deBoor [2, pp. 162-163] comment that such iterative procedures are associated with finding a nonsingular matrix C such that $G = I - C^{-1}A$, and $k = C^{-1}y$. They further comment that the objective in such a procedure is to find a C such that C is easy to invert and G produces a good convergence rate for the above iterative process. This is closely connected with the concept of a splitting $A = M - N$ where A and M are nonsingular $n \times n$ matrices. Splittings (which, as mentioned in Chapter 1, lead to the iterative processes $x^{(m+1)} = (I - M^{-1}A)x^{(m)} + M^{-1}y$) have been considered in detail (see Varga [15], Mangasarian [8], [9]). It appears, however, that the concept of an approximate inverse has not been fully exploited in connection with iterative procedures. The comments in [2] can be extended to approximate inverses. That is, the objective is to find an approximation B to A^{-1} such that B is easy to obtain and $G = I - BA$ produces a good convergence rate in the associated iterative procedure.

The approximate inversion procedures considered in this

thesis are local procedures in the sense that they determine the i'th row in the approximate inverse B to A from entries in A that are "near" the i'th row of A. One application of such techniques, as demonstrated by the experimental results of previous chapters, occurs in connection with least-squares approximation by cubic splines.

5.2. TWO DIMENSIONAL PROBLEMS

The local procedures of Chapter 2 and Chapter 3 lead naturally to the consideration of local two dimensional procedures in Chapter 4. As experimental results indicate, these local techniques for getting an approximate inverse are highly effective for interpolation problems involving the two dimensional spline of Appendix D. Chapter 4 by no means covers the full extent of two dimensional local problems. The success with the two dimensional problem considered in Chapter 4 suggests further experimental work with other two dimensional linear operators (for example operators associated with different two dimensional splines). Further experimental work with non-circulant problems and with various regions in the plane is also suggested. The results with the extended methods of Chapter 3 suggest that similar extensions be studied for two dimensional approximate inverses. An investigation of transform theory for the two dimensional circulant case and an extension of the MMq technique to the two dimensional

circulant problem is further suggested.

APPENDIX A

AN EXCHANGE ALGORITHM FOR THE MMq TECHNIQUE

In this appendix, we give an APL function for finding the elements of MMq (A) when A is an $n \times n$ symmetric band-circulant matrix of band width $2p+1$. Let A have band elements $(a_{-p},\ldots,a_0,\ldots,a_p)$ (with $a_{-j} = a_j$ $1 \le j \le p$). The function MINMAX of Figure A1 has arguments Q and A where $Q = q$ and A is the vector $(a_0,a_1,\ldots,a_p)$ and the output of MINMAX is the vector $(b_0,b_1,\ldots,b_q)$ where MMq (A) has band elements $(b_{-q},\ldots,b_0,\ldots,b_q)$. We include results in Figures A2 and A3 for $1 \le q \le 6$ for the matrices T_2 (the vector SPLINE) and T_4 (the vector INT). The value of R in line [1] of MINMAX determines the number of points in [0,1] on which the exchange algorithm is performed. In Figure A1 we use $2R+1 = 201$ points.

```
      SPLINE
1.0785714   0.53169643   0.053571429   0.00044642857
      1 MINMAX SPLINE
1.6480611   -0.7396419
      2 MINMAX SPLINE
2.0262194   -1.1608087   0.45050771
      3 MINMAX SPLINE
2.1531952   -1.3099026   0.66096314   -0.24971522
      4 MINMAX SPLINE
2.1910973   -1.3552183   0.73012362   -0.3593425
      0.13498772
      5 MINMAX SPLINE
2.2022601   -1.368595   0.75081626   -0.39460662
      0.19307848   -0.072393474
      6 MINMAX SPLINE
2.205239   -1.3722485   0.75670856   -0.4050378
      0.21170311   -0.10346356   0.038798511
```

Figure A2

```
     ∇ B←Q MINMAX A;T;AA;TT;T1;I;M;V;D;CV;U;CC1;D1;CC;
       LI;S;C1;H;TP;SG;P;G;R;NN
[1]    R←100
[2]    T←0,((⍳R)÷R)
[3]    AA←A[1],2×A[1+⍳(⍴A)-1]
[4]    I←1
[5]    V←(R+1)⍴1
[6]   L0:V←V,200((R+1)⍴I)×T
[7]    I←I+1
[8]    →(I≤M←(⍴AA)⌈(NN←Q+1))/L0
[9]    V←((M+1),(R+1))⍴V
[10]   D←AA+.×V[⍳⍴AA;⍳(R+1)]
[11]   T1←T[1,(⍳NN)×⌊(R+1)÷NN]
[12]  LOOP:CV←T⍳T1
[13]   D1←D[CV]
[14]   U←(200(T1∘.×00,⍳NN))×⍉((NN+1),(NN+1))⍴D1
[15]   U[;NN+1]←¯1*⍳NN+1
[16]   CC1←((NN+1)⍴1)⌹U
[17]   CC←CC1[⍳NN]
[18]   LI←1-D×CC+.×V[⍳⍴CC;⍳(R+1)]
[19]   S←LI[C1←(⍋|LI)[1]]
[20]   →((|S)≤(|H←CC1[NN+1]))/L01
[21]   TP←T1,T[C1]
[22]   TP←TP[G←⍋TP]
[23]   SG←(×LI[CV]),(×S)
[24]   SG←SG[G]
[25]   P←(T[C1]=TP)/⍳⍴TP
[26]   →((⍴P)>1)/L01
[27]   →((P=1)∨(P=⍴TP))/LP
[28]   →((SG[P-1]=SG[P]),(SG[P+1]=SG[P]))/L1,L2
[29]  L1:T1←(((P-2)⍴1),0,(((⍴TP)-P-1)⍴1))/TP
[30]   →LOOP
[31]  L2:T1←((P⍴1),0,(((⍴TP)-P+1)⍴1))/TP
[32]   →LOOP
[33]  LP:→((P=1),(P=⍴TP))/L3,L4
[34]  L3:→(SG[1]=SG[2])/L5
[35]   T1←¯1↓TP
[36]   →LOOP
[37]  L5:T1←(1 0 ,(((⍴TP)-2)⍴1))/TP
[38]   →LOOP
[39]  L4:→(SG[(⍴TP)-1]=SG[⍴TP])/L6
[40]   T1←1↓TP
[41]   →LOOP
[42]  L6:T1←((((⍴TP)-2)⍴1), 0 1)/TP
[43]   →LOOP
[44]  L01:B←CC[1],0.5×CC[1+⍳((⍴CC)-1)]
     ∇
```

Figure A1

```
      INT
1   0.25
      1 MINMAX INT
1.1428571  -0.28571429
      2 MINMAX INT
1.15385  -0.30770005  0.076925012
      3 MINMAX INT
1.1546392  -0.30927835  0.082474227  -0.020618557
      4 MINMAX INT
1.1546961  -0.30939227  0.082872928  -0.022099446
      0.0055248619
      5 MINMAX INT
1.1547003  -0.30940061  0.082902126  -0.022205927
      0.0059215805  -0.0014803951
      6 MINMAX INT
1.1547005  -0.30940069  0.082903665  -0.022213673
      0.0059500943  -0.0015867047  0.00039667617
```

Figure A3

APPENDIX B

A PROGRAM FOR FINDING SPECTRAL RADIUS

For the $n \times n$ matrix G we give an APL program in Figure B1 which finds $\rho(G) = \lim_{m\to\infty} (||G^m||_\infty)^{1/m}$. LIMIT must be defined before the program is executed and it represents the allowable deviation between successive approximations to $\rho(G)$. The algorithm uses values of m from the sequence $1,2,4,8,16,\ldots$. We also include in Figure B1 the result of using EIGG on the indicated test matrix.

```
     ∇ R←EIGG G;J;C;L
[1]    R←1+J←L←0
[2]   LO:G←G÷C←⌈/+/|G
[3]    →((|(R←R×C*÷2*J)-L)≤LIMIT)/0
[4]    G←G+.×G
[5]    →LO+0×L←R+0×J←J+1
     ∇


       LIMIT
1E¯8


       TEST
  3   5   6   2
  8   9   6   4
  5   6   7   3
  4   5   6   2


       EIGG TEST
21.05300653
```

Figure B1

APPENDIX C

TEST MATRICES

Four of our test matrices arise in connection with one dimensional spline approximation problems. The piecewise cubic spline defined on $\mathbb{R}$ and with support $[-2,2]$ is given by

$$B(x) = \begin{cases} \frac{1}{4}(x+2)^3, & x \in [-2,-1) \\ \frac{1}{4} + \frac{3}{4}(x+1) + \frac{3}{4}(x+1)^2 - \frac{3}{4}(x+1)^3, & x \in [-1,0) \\ \frac{1}{4} + \frac{3}{4}(1-x) + \frac{3}{4}(1-x)^2 - \frac{3}{4}(1-x)^3, & x \in [0,1) \\ \frac{1}{4}(2-x)^3, & x \in [1,2] \end{cases} \quad (C1)$$

We consider the problem of least-squares approximation of the function $f : [0,1] \to \mathbb{R}$ by a linear combination of the basic splines $S_k(x) = B_{N+1}\left(\frac{x - x_k}{h}\right)$ where $h = \frac{1}{N}$; $x_k = kh$, $-1 \leq k \leq N+1$. We let $g(x) = \sum_{k=-1}^{N+1} a_k S_k(x)$ and seek the a_k's which minimize $\int_0^1 (g(x)-f(x))^2 dx$. This produces the linear system

$$\sum_{k=-1}^{N+1} a_k \langle S_j, S_k \rangle = \langle f, S_j \rangle, \quad -1 \leq j \leq N+1 \quad (C2)$$

where $\langle S_j, S_k \rangle = \int_0^1 S_j(x) S_k(x) dx$ and $\langle f, S_j \rangle = \int_0^1 f(x) S_j(x) dx$. Since $|j-k| > 3$ implies $\langle S_j, S_k \rangle = 0$, a band matrix arises. (For a more detailed consideration of cubic spline approximation problems see Curtis [3], Powell [13].) We let $n = N+3$ and

denote the $n \times n$ matrix associated with the above linear system by A. The 3×3 upper left block of A is given by

$$h \begin{bmatrix} 0.008928571429 & 0.05758928571 & 0.02678571429 \\ 0.05758928571 & 0.5392857143 & 0.4741071429 \\ 0.02678571429 & 0.4741071429 & 1.069642857 \end{bmatrix}$$

The lower right 3×3 block of A is formed from the above array by first interchanging the first and last rows and then interchanging the first and last columns. The matrix A is symmetric and the non-zero row elements from the diagonal out in rows 4 to N-3 are h times

$$1.078571429,\ 0.5316964286,\ 0.05357142857,\ 0.0004464285714\ .$$

Since the factor h occurs in all terms in the left hand side of the equations (C2), we may divide these equations by h and produce the matrix $\frac{1}{h}A$. Our test matrix T_1 is $\frac{1}{h}$ A for $n = 20$. Our test matrix T_2 is the 20×20 band-circulant matrix (Definition 2.1.1) whose fourth row is identical to the fourth row of T_1.

As well as the least-squares approximation problem, we consider a cubic spline interpolation problem. Let $f(x_k) = f_k$, $0 \leq k \leq N$ and let $f'(0) = s_1$, $f'(1) = s_2$ and again let $g(x) = \sum_{k=-1}^{N+1} a_k S_k(x)$. We seek the a_k's such that $f_k = g(x_k)$, $0 \leq k \leq N$ and $g'(0) = s_1$, $g'(1) = s_2$. This gives the linear system

$$
\begin{aligned}
&-\frac{3}{4h}a_{-1} + \frac{3}{4h}a_1 = s_1 \\
&\frac{1}{4}a_{j-1} + a_j + \frac{1}{4}a_{j+1} = f_j,\ 0 \le j \le N \\
&-\frac{3}{4h}a_{N-1} + \frac{3}{4h}a_{N+1} = s_2
\end{aligned}
$$

which gives rise to the $(N+3) \times (N+3)$ matrix (see also Kammerer and Reddien [7])

$$
\begin{bmatrix}
-\frac{3}{4}N & 0 & \frac{3}{4}N & 0 & & & & 0 \\
\frac{1}{4} & 1 & \frac{1}{4} & 0 & . & & & 0 \\
0 & \frac{1}{4} & 1 & \frac{1}{4} & 0 & & & 0 \\
\vdots & & & & & & & \vdots \\
0 & & & & 0 & \frac{1}{4} & 1 & \frac{1}{4} \\
0 & & & & 0 & -\frac{3}{4}N & 0 & \frac{3}{4}N
\end{bmatrix}.
$$

We denote this matrix for the case $N+3 = 20$ by T_3. Our test matrix T_4 is the 20×20 band-circulant matrix whose second row is identical to the second row of T_3.

We denote by T_5 the 20×20 band matrix constructed in the following manner. Let $S = \left\{\frac{1}{20}, \frac{2}{20}, \ldots, \frac{19}{20}, \frac{20}{20}\right\}$ and let $\alpha(s)$ denote an element picked randomly from S. The main diagonal of T_5 has elements of the form $0.6 + 0.6\,\alpha(s)$, the diagonals immediately above and below the main diagonal have elements of the form $0.4 + 0.3\,\alpha(s)$, and the second diagonals

above and below the main diagonal have elements of the form $0.1 + 0.1\,\alpha(S)$. The remaining elements of T_5 are all zeros. In Table C1 we list some experimental results with our test matrices.

Table C1

Experimental results with some of the standard methods of Chapter 1 for the test matrices of Appendix C. G is the iteration matrix in each case and ω_b was found experimentally to ± 0.005. All computational complexities ignore terms in $\frac{1}{n}$ where n is the order of the system under consideration.

Test matrix	Iterative process						
	Jacobi		Gauss-Seidel		successive overrelaxation		
	$\rho(G)$	$E(G)$	$\rho(G)$	$E(G)$	ω_b	$\rho(G)$	$E(G)$
T_1	1.28	diverges	0.900	57	1.460	0.578	13
T_2	1.09	diverges	0.796	26	1.340	0.618	15
T_3	0.526	3.1	0.333	1.8	1.045	0.280	2.4
T_4	0.500	2.9	0.321	1.8	1.075	0.255	2.2
T_5	1.38	diverges	0.720	12	1.210	0.632	11

APPENDIX D

A TWO DIMENSIONAL SPLINE

In Chapter 4, we make use of a two dimensional spline defined on a hexagonal region in the plane. For a detailed description of such splines see Frederickson [4]. In reference to Figure D1, our basic triangular spline has the value $\frac{1}{2}$ at A and the value $\frac{1}{12}$ at each of B,C,D,E,F and G, and vanishes outside the hexagonal region of Figure D1. All the triangles in Figure D1 are equilateral.

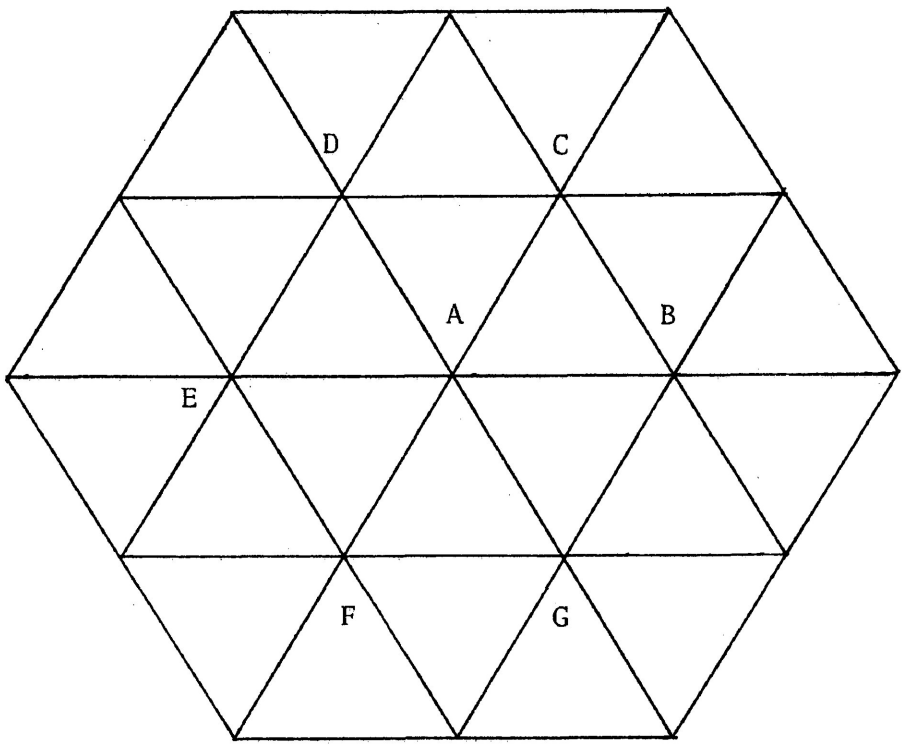

Figure D1

APPENDIX E

A FORTRAN PROGRAM† FOR SPECTRAL RADIUS IN THE TWO DIMENSIONAL CIRCULANT CASE

The subroutine SPECRA of Figure E1 finds the spectral radius of the two dimensional circulant operator represented, in the notation of Chapter 4, by the $m \times n$ array $G = (G_{i,j})$ of $(2p+1) \times (2p+1)$ matrices $G_{i,j}$. This subroutine applies to the cases where $g_{i,j}^{t,u} = g_{i,j}^{-t,-u}$. The argument G in the subroutine is one of the matrices $G_{i,j}$, IG is 2p+1, M is m, N is n, and RAD is the spectral radius determined by the subroutine.

† designed for compilation under WATFIV.

```
      SUBROUTINE SPECRA(G,IG,M,N,RAD)
      DIMENSION G(IG,IG)
      RAD=0.
      IGG=(IG-1)/2+1
      RM=M
      RN=N
      IGG1=IGG+1
      MM=(M+1)/2
      CALL HSPRAD(G,IG,RM,RN,IGG1,MM,M,1,N,RAD)
      CALL HSPRAD(G,IG,RM,RN,IGG1,1,MM,N,N,RAD)
      WRITE(6,100) RAD
100   FORMAT('0','SPECTRAL RADIUS OF I-BA IS  ',E15.8)
      RETURN
      END

      SUBROUTINE HSPRAD(G,IG,RM,RN,IGG1,L11,L12,L21,L22,RAD)
      DIMENSION G(IG,IG)
      PI2=6.2831853
      DO 1 K1=L11,L12
      DO 1 K2=L21,L22
      RK1=K1
      RK2=K2
      IGG=IGG1-1
      S=G(IGG,IGG)
      SS=0.
      DO 2 IR=IGG1,IG
      RIR=IR-IGG
2     SS=SS+G(IR,IGG)*COS(PI2*RK1*RIR/RM)
      S=S+2*SS
      SSS=0.
      DO 3 IR=1,IG
      DO 3 IS=IGG1,IG
      RIR=IR-IGG
      RIS=IS-IGG
3     SSS=SSS+G(IR,IS)*COS(PI2*(RK1*RIR/RM+RK2*RIS/RN))
      S=S+2*SSS
      S=ABS(S)
      IF(S.LT.RAD) GO TO 1
      RAD=S
1     CONTINUE
      RETURN
      END
```

Figure E1

APPENDIX F

FORTRAN PROGRAMS† FOR FINDING THE LSq AND DBq APPROXIMATE INVERSES IN THE TWO DIMENSIONAL CIRCULANT CASE

In Figure F1, we list a FORTRAN program for finding DBq (A).

```
      SUBROUTINE DBQ(A,B,G,RH,IA,IB,IG)
      DIMENSION A(IA,IA),B(IB,IB),G(IG,IG),RH(IB,IB)
      WRITE(6,102)
102   FORMAT('0','THE OPERATOR A')
      CALL OUTPUT(A,IA,IA,1,1)
      DO 6 I=1,IB
      DO 6 J=1,IB
6     RH(I,J)=0.
      IRH=(IB-1)/2+1
      RH(IRH,IRH)=1.
      W=1.
      CALL SOR(A,B,RH,IA,IB,W)
      WRITE(6,100)
100   FORMAT('0','THE OPERATOR B')
      CALL OUTPUT(B,IB,IB,1,1)
      CALL MULT(B,A,G,IA,IB,IG)
      DO 1 I=1,IG
      DO 1 J=1,IG
1     G(I,J)=-G(I,J)
      IGG=(IG-1)/2+1
      G(IGG,IGG)=1+G(IGG,IGG)
      WRITE(6,101)
101   FORMAT('0','THE OPERATOR G IS')
      CALL OUTPUT(G,IG,IG,1,1)
      RETURN
      END
```

Figure F1

The argument A is an element of the array defining the circulant operator whose approximate inverse is being determined, and the argument B is an element of that inverse. The argument G is an element of the iteration operator associated with the diagonal block approximate inverse. The argument RH is

† designed for compilation under WATFIV

used to create the right hand side of the linear system associated with the diagonal block approximate inversion technique. The arguments IA, IB, and IG are the dimensions of A, B, and G respectively. The subroutines MULT nad OUTPUT are described in Appendix G.

In Figure F2, we list a FORTRAN program for finding LSq (A). The arguments A,B,G, IA, IB, IG are the same as above. The arguments ATA, RHSQ, and HATA are matrices created and used in the subroutine LSq. IHATA and IATA are the dimensions of HATA and ATA respectively and they are defined by IHATA = 3*IA - 2 and IATA = 2*IA - 1 in the calling program.

Figure F3 contains a subroutine to solve by successive overrelaxation the linear systems created by DBQ and LSQ. Our LSQ and DBQ subroutines use a relaxation factor of 1, however this is easily modified.

```
      SUBROUTINE LSQ(A,B,G,ATA,RHSQ,HATA,IHATA,IA,IB,IG,IATA)
      DIMENSION A(IA,IA),B(IB,IB),G(IG,IG)
      DIMENSION ATA(IATA,IATA),RHSQ(IB,IB)
      DIMENSION HATA(IHATA,IHATA)
      IA1=2*IA-1
      DO 8 I=1,IB
      DO 8 J=1,IB
8     RHSQ(I,J)=0.
      DO 9 I=1,IHATA
      DO 9 J=1,IHATA
9     HATA(I,J)=0.
      DO 5 I=IA,IA1
      DO 5 J=IA,IA1
5     HATA(I,J)=A(I-IA+1,J-IA+1)
      DO 1 I=1,IATA
      DO 1 J=1,IATA
      S=0.
      DO 2 K=1,IA
      DO 2 L=1,IA
2     S=S+A(K,L)*HATA(I-1+K,J-1+L)
1     ATA(I,J)=S
      LT=(IA-1)/2+1
      IBT=(IB-1)/2+1
      DO 3 I=1,IB
      DO 3 J=1,IB
      I1=IBT-I+LT
      I2=IBT-J+LT
      IF(I1.LT.1.OR.I1.GT.IA.OR.I2.LT.1.OR.I2.GT.IA) GO TO 3
      RHSQ(I,J)=A(I1,I2)
3     CONTINUE
      W=1.
      WRITE(6,102)
102   FORMAT('0','THE OPERATOR A')
      CALL OUTPUT(A,IA,IA,1,1)
      CALL SOR(ATA,B,RHSQ,IATA,IB,W)
      WRITE(6,100)
100   FORMAT('0','THE OPERATOR B')
      CALL OUTPUT(B,IB,IB,1,1)
      CALL MULT(B,A,G,IA,IB,IG)
      DO 4 I=1,IG
      DO 4 J=1,IG
4     G(I,J)=-G(I,J)
      IGG=(IG-1)/2+1
      G(IGG,IGG)=1+G(IGG,IGG)
      WRITE(6,101)
101   FORMAT('0','THE OPERATOR G')
      CALL OUTPUT(G,IG,IG,1,1)
      RETURN
      END
```

Figure F2

```
      SUBROUTINE SOR(A,B,RH,IA,IB,W)
      INTEGER CTR/1/
      DIMENSION A(IA,IA),B(IB,IB),RH(IB,IB)
      DIMENSION X1(13,13),AM(10,10),RHM(13,13)
      REAL X1/169*0./
      DO 30 I=1,IB
      DO 30 J=1,IB
30    B(I,J)=0.
      LT=(IA-1)/2+1
      DO 10 I=1,IA
      DO 10 J=1,IA
10    AM(I,J)=-A(I,J)/A(LT,LT)
      AM(LT,LT)=0.
      DO 11 I=1,IB
      DO 11 J=1,IB
11    RHM(I,J)=RH(I,J)/A(LT,LT)
50    DO 1 J=1,IB
      DO 1 I=1,IB
      S=0.
      DO 2 K=1,IB
      DO 2 L=1,IB
      I1=LT+I-K
      I2=LT+J-L
      IF(I1.LT.1.OR.I1.GT.IA.OR.I2.LT.1.OR.I2.GT.IA) GO TO 2
      S=S+B(K,L)*AM(I1,I2)
2     CONTINUE
      B(I,J)=(S+RHM(I,J))*W+B(I,J)*(1-W)
1     CONTINUE
      T=0.
      DO 20 I=1,IB
      DO 20 J=1,IB
      TT=ABS(X1(I,J)-B(I,J))
      IF(TT.LE.T) GO TO 20
      T=TT
20    CONTINUE
      IF(T.LE.1.0E-6) GO TO 60
      DO 40 I=1,IB
      DO 40 J=1,IB
40    X1(I,J)=B(I,J)
      CTR=CTR+1
      IF(CTR.LE.100) GO TO 50
60    WRITE(6,500) CTR
500   FORMAT('0','NO. OF ITERATIONS FOR S.O.R. IS',I3)
      RETURN
      END
```

Figure F3

APPENDIX G

FORTRAN PROGRAMS† FOR TWO DIMENSIONAL ITERATIVE PROCESSES IN THE CIRCULANT CASE

This appendix contains FORTRAN programs for performing iterative processes based on approximate inverses in the two dimensional circulant case. Figure G1 contains the main program. In this example the LS_1 approximate inversion technique is being employed to solve iteratively the problem represented in Table 4.6.1. Very little modification is required to employ the diagonal block technique.

In the next few figures we list the subroutines used in connection with our two dimensional iterative processes. Figure G2 contains the subroutine EXTEND which performs the periodic extension of a two dimensional array according to the description associated with (4.5.1).

The subroutine MULT of Figure G3 determines B A according to (4.5.2), and the subroutine LINOP of Figure G3 finds A X according to (4.5.1).

The subroutine ITERAT of Figure G4 performs the iteration $X^{(p+1)} = G\,X^{(p)} + BY$, and the subroutine MAXAB of Figure G4 determines $\delta^{(p)}$ according to (4.6.3).

† designed for compilation under WATFIV.

```
C       TWO DIMENSIONAL ITERATIVE PROCESSES
        INTEGER P
        DIMENSION X1(29,39),X2(29,39),BY(29,39)
        DIMENSION A(3,3),B(3,3),G(5,5),RH(3,3)
        DIMENSION ATA(7,7),IHATA(7,7),RHSQ(7,7)
        EPSIL=1.0E-7
        MP=29
        NP=39
        IA=3
        IB=3
        IG=5
        IHATA=3*IA-2
        IATA=2*IA-1
        P=(IG-1)/2
        LT=P+1
        MP1=MP-P
        NP1=NP-P
        CALL TEST(X1,MP,NP,P)
        CALL EXTEND(X1,MP,NP,P)
        WRITE(6,450)
450     FORMAT('0','THE ARRAY Y IS')
        CALL OUTPUT(X1,MP,NP,P,0)
        READ,((A(I,J),I=1,IA),J=1,IA)
        CALL LSQ(A,B,G,ATA,RHSQ,IHATA,IHATA,IA,IB,IG,IATA)
        CALL LINOP(X1,BY,B,MP,NP,IB)
800     CALL ITERAT(BY,X1,X2,G,MP,NP,IG)
        ICC=2
        CALL MAXAB(X1,X2,MP,NP,P,S)
        WRITE(6,400) S
400     FORMAT('0','SUP NORM (X(M+1)-X(M))=   ',E15.8)
        IF(S.LT.EPSIL) GO TO 900
        CALL ITERAT(BY,X2,X1,G,MP,NP,IG)
        ICC=1
        CALL MAXAB(X1,X2,MP,NP,P,S)
        WRITE (6,400) S
        IF(S.LT.EPSIL) GO TO 900
        GO TO 800
900     WRITE(6,700)
700     FORMAT('0','SOLUTION IS')
        CALL TEST(BY,MP,NP,P)
        GO TO (35,36),ICC
35      CALL OUTPUT(X1,MP,NP,P,0)
        CALL EXTEND(X1,MP,NP,P)
        CALL LINOP(X1,X2,A,MP,NP,IA)
        CALL MAXAB(BY,X2,MP,NP,P,S)
        WRITE(6,703) S
703     FORMAT('0','MAX ERROR IN TESTED PRODUCT IS  ',E15.8)
        CALL SPECRA(G,IG,MP1-P,NP1-P,RAD)
        GO TO 38
36      CALL OUTPUT(X2,MP,NP,P,0)
        CALL EXTEND(X2,MP,NP,P)
        CALL LINOP(X2,X1,A,MP,NP,IA)
        CALL MAXAB(BY,X1,MP,NP,P,S)
        WRITE(6,703) S
        CALL SPECRA(G,IG,MP1-P,NP1-P,RAD)
38      STOP
        END
```

Figure G1

```
      SUBROUTINE EXTEND(X,MP,NP,P)
      DIMENSION X(MP,NP)
      INTEGER P
      M=MP-2*P
      N=NP-2*P
      JR1=MP-P
      JR2=JR1+1
      LT=P+1
      IB1=MP-P
      IB2=IB1+1
      DO 1 I=1,P
      DO 1 J=1,P
1     X(I,J)=X(I+M,J+N)
      DO 2 I=1,P
      DO 2 J=LT,JR1
2     X(I,J)=X(I+M,J)
      DO 3 I=1,P
      DO 3 J=JR2,NP
3     X(I,J)=X(I+M,J-N)
      DO 4 I=LT,IB1
      DO 4 J=JR2,NP
4     X(I,J)=X(I,J-N)
      DO 5 I=IB2,MP
      DO 5 J=JR2,NP
5     X(I,J)=X(I-M,J-N)
      DO 6 I=IB2,MP
      DO 6 J=LT,JR1
6     X(I,J)=X(I-M,J)
      DO 7 I=IB2,MP
      DO 7 J=1,P
7     X(I,J)=X(I-M,J+N)
      DO 8 I=LT,IB1
      DO 8 J=1,P
8     X(I,J)=X(I,J+N)
      RETURN
      END
```

Figure G2

```
      SUBROUTINE MULT(B,A,C,IA,IB,IC)
      DIMENSION A(IA,IA),B(IB,IB),C(IC,IC)
      DO 10 I=1,IC
      DO 10 J=1,IC
10    C(I,J)=0.
      LT=(IA-1)/2+1
      MR=IC-LT+1
      DO 1 I=LT,MR
      DO 1 J=LT,MR
      DO 1 K=1,IA
      DO 1 L=1,IA
1     C(I-LT+K,J-LT+L)=C(I-LT+K,J-LT+L)+B(I-LT+1,J-LT+1)*A(K,L)
      RETURN
      END

      SUBROUTINE LINCP(X,Y,A,MP,NP,IA)
      DIMENSION A(IA,IA),X(MP,NP),Y(MP,NP)
      INTEGER P,P1
      P=(IA-1)/2
      P1=P+1
      I2=MP-P
      J2=NP-P
      DO 1 I= P1,I2
      DO 1 J=P1,J2
      S=0.
      DO 2 K=1,IA
      DO 2 L=1,IA
2     S=S+A(K,L)*X(I+K-P1,J+L-P1)
1     Y(I,J)=S
      RETURN
      END
```

Figure G3

```
      SUBROUTINE ITERAT (BY,X1,X2,G,MP,NP,IG)
      DIMENSION BY(MP,NP),X1(MP,NP),X2(MP,NP),G(IG,IG)
      INTEGER P
      P=(IG-1)/2
      LT=P+1
      IB=MP-P
      JR=NP-P
      CALL EXTEND(X1,MP,NP,P)
      CALL LINOP(X1,X2,G,MP,NP,IG)
      DO 1 I=LT,IB
      DO 1 J=LT,JR
1     X2(I,J)=BY(I,J)+X2(I,J)
      RETURN
      END

      SUBROUTINE MAXAB(X,Y,MP,NP,P,S)
      DIMENSION X(MP,NP),Y(MP,NP)
      INTEGER P
      JR=NP-P
      IB=MP-P
      LT=P+1
      S=ABS(X(LT,LT)-Y(LT,LT))
      DO 1 I=LT,IB
      DO 1 J=LT,JR
      T=ABS(X(I,J)-Y(I,J))
      IF(T.LE.S) GO TO 1
      S=T
1     CONTINUE
      RETURN
      END
```

Figure G4

The subroutine TEST of Figure G5 determines, according to statement 1, the elements of Y for testing an iterative process on $Ax = Y$.

```
      SUBROUTINE TEST(Y,MP,NP,P)
      DIMENSION Y(MP,NP)
      INTEGER P
      JR=NP-P
      IB=MP-P
      LT=P+1
      RN=NP-2*P
      RM=MP-2*P
      PI=3.141592 65
      DO 1 I=LT,IB
      RI=I-P
      DO 1 J=LT,JR
      RJ=J-P
1     Y(I,J)=SIN(2*PI*RI/RM)*SIN(2*PI*RJ/RN)
      RETURN
      END
```

Figure G5

Finally Figure G6 contains an output subroutine for printing either on $m \times n$ matrix or the matrix X given $E(X)$ where $E(X)$ was defined in Section 4.5. The program of Figure G1 also uses subroutines from Appendix E and Appendix F.

```
      SUBROUTINE OUTPUT(X,MP,NP,P,MOD)
      DIMENSION X(MP,NP)
      INTEGER P,PR
      PR=NP-P
      IF(MOD.EQ.1) GO TO 400
      N=NP-2*P
      M3=P+1
      M4=MP-P
      GO TO 401
400   M3=1
      M4=MP
      N=NP
401   JJ=1
      II=N/10
502   M1=(JJ-1)*10+M3
      M2=JJ*10+M3-1
      IF (JJ.GT.II) GO TO 504
      WRITE(6,503)
      WRITE(6,610) ((X(I,J),J=M1,M2),I=M3,M4)
      IF(M2.EQ.NP.OR.(M2.EQ.PR.AND.MOD.NE.1)) GO TO 501
      JJ=JJ+1
      GO TO 502
504   M2=N+M3-1
      WRITE(6,503)
      IND=M2-M1+1
      GO TO(31,32,33,34,35,36,37,38,39),IND
31    WRITE(6,601) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
32    WRITE(6,602) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
33    WRITE(6,603) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
34    WRITE(6,604) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
35    WRITE(6,605) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
36    WRITE(6,606) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
37    WRITE(6,607) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
38    WRITE(6,608) ((X(I,J),J=M1,M2),I=M3,M4)
      GO TO 501
39    WRITE(6,609) ((X(I,J),J=M1,M2),I=M3,M4)
601   FORMAT('0',E12.4)
602   FORMAT('0',2E12.4)
603   FORMAT('0',3E12.4)
604   FORMAT('0',4E12.4)
605   FORMAT('0',5E12.4)
606   FORMAT('0',6E12.4)
607   FORMAT('0',7E12.4)
608   FORMAT('0',8E12.4)
609   FORMAT('0',9E12.4)
610   FORMAT('0',10E12.4)
503   FORMAT('-')
501   RETURN
      END
```

Figure G6

BIBLIOGRAPHY

[1] Charmonman, S. and Julius, R.S., Explicit inverses and condition numbers of certain circulants, Math. of Comp., 22 (1968), 428-430.

[2] Conte, S.D. and deBoor, Carl, Elementary Numerical Analysis, An Algorithmic Approach, Second Edition, McGraw-Hill, New York, (1972).

[3] Curtis, A.R., The approximation of a function of one variable by cubic splines, Numerical Approximation to Functions and Data, J.G. Hayes, Ed., University of London, The Athlone Press, London, (1970), 28-42.

[4] Frederickson, P.O., Generalized triangular splines, Lakehead University Math. Report #7-71, (1971).

[5] Frederickson, P.O., Quasi-interpolation, extrapolation, and approximation on the plane, Proc. First Man. Conf. on Num. Maths, Utilitas Math., Winnipeg (1971), 159-167.

[6] Hoskins, W.D. and Ponzo, P.J., Some properties of a class of band matrices, Math. of Comp., 26 (1972), 393-400.

[7] Kammerer, W.J. and Reddien, G.W. Jr., Local convergence of smooth cubic spline interpolates, Siam J. Numer. Anal., 9 (1972), 687-694.

[8] Mangasarian, O.L., A convergent splitting of matrices, Numer. Math., 15 (1970), 351-353.

[9] Mangasarian, O.L., Convergent generalized monotone splitting of matrices, Math. of Comp., 25 (1971), 649-653.

[10] Marek, I., On the row sum criterion and the convergence of some iterative processes, Numer. Math., 13 (1969), 207-216.

[11] Marek, I. and Varga, R.S., Nested bounds for the spectral radius, Numer. Math., 14 (1969), 49-70.

[12] Meinardus, Günter, Approximation of Functions: Theory and Numerical Methods, Springer Tracts in Natural Philosophy, Vol. 13, Springer-Verlag New York Inc. (1967) (Translated by L. Schumaker.)

[13] Powell, M.J.D., The local dependence of least squares cubic splines, Siam J. Numer. Anal., 6 (1969), 398-413.

[14] Varga, Richard S., Eigenvalues of circulant matrices, Pacific J. of Math., 4 (1954), 151-160.

[15] Varga, Richard S., Matrix Iterative Analysis, Prentice-Hall, Inc., Englewood Cliffs, N.J. (1962).

[16] Young, David M., Iterative Solution of Large Linear Systems, Academic Press, Inc., New York (1971).

[17] Young, David M., On the solution of large systems of linear algebraic equations with sparse, positive definite matrices, Centre for Numerical Analysis, The Univ. of Texas at Austin, Tech. Rep. CNA 55, (1972).

[18] Young, David M., On the consistency of linear stationary iterative methods, Siam J. Numer. Anal., 9 (1972), 89-96.