

# Technical Report

TR-2006-013

**Kronecker Product Approximations for Dense Block Toeplitz-plus-Hankel  
Matrices**

by

Misha Kilmer, James Nagy

**MATHEMATICS AND COMPUTER SCIENCE**

**EMORY UNIVERSITY**

# Kronecker Product Approximations for Dense Block Toeplitz-plus-Hankel Matrices

Misha Kilmer\* and James Nagy†

September 27, 2006

## Abstract

In this paper, we consider the approximation of dense block Toeplitz-plus-Hankel matrices by sums of Kronecker products. We present an algorithm for efficiently computing the matrix approximation that requires the factorization of matrices of much smaller dimension than that of the original. The main results are described for block Toeplitz matrices with Toeplitz-plus-Hankel blocks (BTTHB), but the algorithms can be readily adjusted for other related structures that arise in image processing applications, such as block Toeplitz with Toeplitz blocks (BTTB) and block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB). Our work extends the techniques in [11, 15], which consider similar matrices, but with the added restriction that the matrices have a banded/block-banded structure. We illustrate the effectiveness of our algorithm by using the output of the algorithm to construct preconditioners for systems from two different applications: diffuse optical tomography and atmospheric image deblurring.

## 1 Introduction

We are interested in the solution of large-scale problems of the form

$$g = Kf, \tag{1}$$

or the regularized solution of discrete ill-posed problems of the form

$$g = Kf + e \tag{2}$$

where the matrix  $K$  has special structure. Here,  $e$  represents unknown noise within the measured signal. Typically, the matrix  $K$  represents the discretization of a first kind Fredholm integral operator, and therefore will be ill-conditioned with no gap in its singular value spectrum. Regularization is needed to solve the discrete ill-posed problem (2) because, otherwise, when forming the standard least-squares solution the noise in the data is magnified by the small singular values, thereby rendering the solution worthless. In either case of (1) or (2), we are interested in constructing approximations to  $K$  that are relatively quickly and easily inverted for the purpose of either preconditioning the system or computing approximate solutions directly.

Efficient construction of these approximations is obtained by exploiting structure in  $K$ . In this paper we focus on Toeplitz and Hankel structures that arise naturally in many image processing applications (see, for example, [9, 3, 18, 17]). In particular, we consider matrices  $K$  that are block Toeplitz with Toeplitz blocks (BTTB), block Toeplitz with Toeplitz-plus-Hankel blocks (BTTHB), and block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB). In image processing, the determination of whether the blocks are Toeplitz or Toeplitz-plus-Hankel depends on the boundary conditions that are imposed [15]. BTTHB matrices can also occur, for example, in applications where the matrix is related to the discretization of a Green's function (integral equation kernel) for a partial differential equation with appropriate boundary conditions. In fact, our motivation for considering BTTHB matrices stems from the (3D) modeling of a forward problem in diffuse optical tomography [12] where the matrix is related to the

---

\*Department of Mathematics, Tufts University, Misha.Kilmer@tufts.edu. This work funded by NSF grants 0139968 and 0208548.

†Department of Mathematics and Computer Science, Emory University, nagy@mathcs.emory.edu. Research supported in part by the NSF under grant DMS-05-11454 and by an Emory University Research Committee grant.

discretization of the Green’s function for the diffusion equation using an extrapolated boundary condition (see [10] and references therein) as implemented in [2]. Although the two applications mentioned above correspond to matrices arising from discretizations of integral equations, matrices with similar structures also arise as discretizations of partial differential equations.

It has been shown (cf. [11, 15]) that BTTB and BTHTHB matrices arising in certain image processing applications can be decomposed, very efficiently, in terms of a sum of Kronecker products of matrices. This previous work, though, considers problems arising from integral equations where the kernel has compact support. This means that the Toeplitz components of  $K$  are block banded with banded blocks, and that the Hankel parts can only have components in the outermost anti-bands (c.f. [15]). The purpose of this paper is to extend the results and algorithms in [11, 15] to densely structured matrices, and, in particular, to lift all of the imposed restrictions on the Toeplitz and Hankel components of  $K$ . We focus on BTHTHB matrices, and show how to adapt the algorithms to the special case of BTTB matrices, or the more general case of BTHTHB matrices.

This paper is organized as follows. In Section 2, we give definitions and notation, as well as introduce the optimization problem we wish to solve in order to obtain our approximation. At the beginning of Section 3, we introduce some new terminology and give a small scale example of how the new terminology can be used to conveniently express Toeplitz and Hankel matrices. We devote the remainder of Section 3 to reformulating the optimization problem in this new framework. We present our general form algorithm in Section 4 and point out a special case of the algorithm when the Hankel blocks are zero (that is, the BTTB case), and how the algorithm can be generalized to BTHTHB matrices. In Section 5, we give illustrations of how the output of our algorithm can be used as preconditioners for linear systems arising in nonlinear diffuse optical tomographic imaging and in image deblurring. Conclusions and future work are the subject of Section 6.

## 2 Background and Notation

A Toeplitz matrix is one that is constant along its diagonals. Therefore, all the entries in a Toeplitz matrix are uniquely determined by the first row and column of the matrix. Another way of saying this is that the  $i, j$  entry of a Toeplitz matrix  $T$  is given by  $t_{i-j}$ .

Recall that a Hankel matrix is one that is constant along anti-diagonals. Therefore a Hankel matrix can be determined by its first column and last row. In other words, if  $H$  denotes an  $n \times n$  Hankel matrix, then its  $(i, j)$  entry will be defined by value in a list  $h$  of length  $2n - 1$ , namely  $h_{i+j-n-1}$ . Note that a Hankel matrix is always symmetric. If a Hankel matrix is “anti-symmetric” as well, then the  $(i, j)$  entry of  $H$  will be  $h_{|i+j-n-1|}$ .

A Toeplitz-plus-Hankel matrix is simply the sum of a Toeplitz and a Hankel matrix.

A block Toeplitz matrix  $K$  having blocks  $T_i$  can be written as

$$K = \begin{bmatrix} T_0 & T_{-1} & T_{-2} & \cdots & \cdots & T_{1-n_a} \\ T_1 & T_0 & T_{-1} & T_{-2} & \cdots & T_{2-n_a} \\ T_2 & T_1 & T_0 & T_{-1} & \ddots & T_{3-n_a} \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ T_{n_a-1} & \cdots & \cdots & T_2 & T_1 & T_0 \end{bmatrix}.$$

We assume throughout the paper that each block  $T_i$  is  $n_b \times n_b$  and that there are  $n_a$  block rows and  $n_a$  block columns. Thus, the matrix  $K$  is  $n_a n_b \times n_a n_b$ .

We utilize the following notation from [19]:

$$\tilde{T} = \begin{bmatrix} \text{vec}(T_0)^T \\ \vdots \\ \text{vec}(T_{n_a-1})^T \\ \hline \text{vec}(T_{-1})^T \\ \vdots \\ \text{vec}(T_{n_a-2})^T \\ \hline \vdots \\ \vdots \\ \hline \text{vec}(T_{1-n_a})^T \\ \vdots \\ \vdots \\ \hline \text{vec}(T_0)^T \end{bmatrix}, \quad (3)$$

where  $\text{vec}(A)$  unstacks matrix  $A$  by columns to create a column vector. We can think of  $\tilde{T}$  as a block matrix, partitioned by the horizontal lines in (3). Each block of  $\tilde{T}$  has  $n_a$  rows and there are  $n_a$  such block rows. The number of columns of  $\tilde{T}$  is equal to the total number of entries in each  $T_i$ . Therefore,  $\tilde{T}$  is  $n_a^2 \times n_b^2$ .

Following MATLAB convention, we introduce a few more notational conveniences. If  $r$  and  $c$  are vectors of length  $n$ ,

$$\text{toeplitz}(c, r)$$

will create a Toeplitz  $n \times n$  matrix whose first row is  $r$  and last column is  $c$ . Similarly, the command

$$\text{hankel}(c, r)$$

will create a Hankel matrix whose first column is given by  $c$  and last row is given by  $r$ . The notation  $A(:, j)$  refers to the  $j$ th column of matrix  $A$  while the notation  $A(j, :)$  refers to the  $j$ th row of  $A$ . The definition  $\text{zeros}(m, n)$  is an  $m \times n$  matrix of all zeros. Finally, the notation  $\text{fliplr}(v)$  where  $v$  is a row vector reverses the order of the entries in the vector  $v$ . As in MATLAB, we can create a matrix using semi-colon notation; for example, if  $v$  and  $w$  are two row vectors of the same dimension,  $[v; w]$  refers to the matrix with 2 rows, the first row comprised of  $v$ , the second comprised of  $w$ . Finally, the notation  $\text{diag}(l_1, l_2, \dots, l_k)$  is a diagonal (or block-diagonal, depending on the size of the  $l_i$ ) matrix with elements  $l_1, \dots, l_k$  on the diagonal.

We already know from [19] that for a predetermined rank  $s$ ,

$$\min_{A_k, B_k} \|K - \sum_{k=1}^s (A_k \otimes B_k)\|_F = \min_{\tilde{a}_k, \tilde{b}_k} \|\tilde{T} - \sum_{k=1}^s (\tilde{a}_k \tilde{b}_k^T)\|_F, \quad (4)$$

where  $\tilde{a}_k = \text{vec}(A_k)$ ,  $\tilde{b}_k = \text{vec}(B_k)$ .

Our goal is to minimize the problem above *subject to* finding Toeplitz  $A_k$  and Toeplitz-plus-Hankel  $B_k = B_k^{(t)} + B_k^{(h)}$ , where  $B_k^{(t)}$  is Toeplitz and  $B_k^{(h)}$  is Hankel. Equivalently, we must find  $\tilde{a}_k, \tilde{b}_k = \tilde{b}_k^{(t)} + \tilde{b}_k^{(h)}$  corresponding to those matrices. We want to do this with as few computations as possible. We will show it is possible to compute this information by computing the SVD of a particular matrix of small dimension as compared to the dimension of  $K$ . Finally, we will show how to use the output of these algorithms to construct preconditioners for certain applications.

### 3 Reformulating the Problem

In our BTTHB matrix, each  $n_b \times n_b$  block is  $T_k = T_k^{(t)} + T_k^{(h)}$  where  $T_k^{(t)}$  is Toeplitz and  $T_k^{(h)}$  is Hankel. There are  $n_a$  block rows and  $n_a$  block columns in  $K$ , so that  $K$  has dimension  $n_a n_b \times n_a n_b$ .

Let us introduce some matrices that will allow us to appropriately characterize the structure in the matrices.

### 3.1 Notation and Example

Let  $Z_n$  be the “downshift” matrix defined by

$$Z_n = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \ddots & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 & 0 \end{bmatrix},$$

that is, a Toeplitz matrix with first row equal to  $e_2^T$  and first column of all zeros. Notice that  $Z^T$  is an “upshift” matrix.

Let  $W_n = [I_n, 0]$  denote the  $n \times (2n - 1)$  matrix where  $I_n$  is the  $n \times n$  identity matrix and the zero block has  $n - 1$  columns.

The purpose of introducing these matrices is to help us view the matrices  $T_k^{(t)}, T_k^{(h)}$  and  $\tilde{T}$  in a particular way. To illustrate the point, consider the  $3 \times 3$  Toeplitz matrix

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 3 \end{bmatrix}.$$

Let  $x = [5, 4, 3, 2, 1]^T$ , noting that the first 3 entries of  $x$  are the entries of the first row, in reverse order, and the remaining 2 entries are the entries in the first column. Define the  $3 \times 5$  matrix  $W_3 = [I_3, 0]$ . Then the Toeplitz matrix can be described by columns as

$$W_3[Z_5^2 x, Z_5^1 x, Z_5^0 x].$$

Next, consider the  $3 \times 3$  Hankel matrix

$$\begin{bmatrix} 7 & 8 & 9 \\ 8 & 9 & 10 \\ 9 & 10 & 11 \end{bmatrix}.$$

Let  $y = [7, 8, 9, 10, 11]^T$ , where the first 3 entries are the entries of the first column and the remaining 2 entries are the last 2 entries from the last row. Then the Hankel matrix is generated as

$$W_3[Z_5^0 y, Z_5^1 y, Z_5^2 y].$$

Let  $E$  denote the  $3 \times 3$  Toeplitz-plus-Hankel matrix obtained by summing the two  $3 \times 3$  matrices above. We define  $p^T = [x^T, y^T]$ . Notice that  $x$  and  $y$  are both of length  $2(3) - 1$ , so  $p$  is of length  $4(3) - 2$ . We then observe that

$$\text{vec}(E) = \begin{bmatrix} [W_3 Z_5^2 | W_3 Z_5^0] & 0 & 0 \\ 0 & [W_3 Z_5^1 | W_3 Z_5^1] & 0 \\ 0 & 0 & [W_3 Z_5^0 | W_3 Z_5^2] \end{bmatrix} \begin{bmatrix} p \\ p \\ p \end{bmatrix},$$

from which it follows that

$$\text{vec}(E)^T = \begin{bmatrix} p^T & p^T & p^T \end{bmatrix} \begin{bmatrix} \begin{bmatrix} (Z_5^2)^T W_3^T \\ (Z_5^0)^T W_3^T \end{bmatrix} & 0 & 0 \\ 0 & \begin{bmatrix} (Z_5^1)^T W_3^T \\ (Z_5^1)^T W_3^T \end{bmatrix} & 0 \\ 0 & 0 & \begin{bmatrix} (Z_5^0)^T W_3^T \\ (Z_5^2)^T W_3^T \end{bmatrix} \end{bmatrix}.$$

### 3.2 Rewriting $\tilde{T} = \sum_{k=1}^s \tilde{a}_k \tilde{b}_k^T$

With that example in mind, we wish to derive an expression for  $\tilde{T}$  in (3). First, we consider how we might rewrite each block of  $\tilde{T}$ , where a block is defined in (3) by use of the horizontal lines. Notice that although  $\tilde{T}$  has  $n_a^2$  rows, *there are only  $2n_a - 1$  distinct rows*: the first  $n_a$  rows of  $\tilde{T}$ , along with the first row of each of the remaining blocks. This follows from the fact that  $T$  is defined completely in terms of

its first (block) row and first (block) column and from the definition of  $\tilde{T}$ . For this reason, we put these  $2n_a - 1$  rows into a matrix  $F$  according to

$$F \equiv \begin{bmatrix} \text{vec}(T_{1-n_a})^T \\ \text{vec}(T_{2-n_a})^T \\ \vdots \\ \text{vec}(T_0)^T \\ \vdots \\ \text{vec}(T_{n_a-1})^T \end{bmatrix} \in \mathbb{R}^{(2n_a-1) \times n_b^2}.$$

Then from equation (3) and with the help of our new notation,

$$\tilde{T} = \begin{bmatrix} W_{n_a} Z_{2n_a-1}^{n_a-1} F \\ W_{n_a} Z_{2n_a-1}^{n_a-2} F \\ \vdots \\ W_{n_a} Z_{2n_a-1}^0 F \end{bmatrix} \in \mathbb{R}^{n_a^2 \times n_b^2}, \quad (5)$$

where  $Z_{2n_a-1}$  is  $(2n_a - 1) \times (2n_a - 1)$  downshift matrix and  $W_{n_a}$  is the  $n_a \times (2n_a - 1)$  matrix  $[I_{n_a}, 0]$ .

Now every row in  $F$  is the transpose of the vectorized form of an  $n_b \times n_b$  Toeplitz-plus-Hankel matrix  $T_k = T_k^{(t)} + T_k^{(h)}$ . Recall that the entries of  $T_k^{(t)}$  are determined once we know its first row and column and the entries of  $T_k^{(h)}$  are determined once we know its first column and last row. Analogous to our example in the preceding subsection, let  $x_k$  be the  $2n_b - 1$  length column vector that contains the entries in the first row, in reverse order, of  $T_k^{(t)}$  followed by the remainder of the terms in the first column. Let  $y_k$  denote the vector containing the first column of  $T_k^{(h)}$  followed by the remaining entries in the last row. For each  $k$ ,  $k = -n_a + 1, \dots, n_a - 1$ , let  $p_k^T = [x_k^T, y_k^T]$  be a row vector of length  $4n_b - 2$ . It follows that

$$\text{vec}(T_k)^T = [p_k^T, p_k^T, \dots, p_k^T] D_R^T, \quad k = -n_a + 1, \dots, n_a - 1,$$

where  $D_R^T$  is a diagonal matrix with  $m$ th block

$$\begin{bmatrix} (Z_{2n_b-1}^{n_b-m-1})^T W_{n_b}^T \\ (Z_{2n_b-1}^m)^T W_{n_b}^T \end{bmatrix},$$

for  $m = 0, \dots, n_b - 1$ .

We define the matrix  $P$  by rows as

$$P = \begin{bmatrix} p_{1-n_a}^T \\ p_{2-n_a}^T \\ \vdots \\ p_0^T \\ \vdots \\ p_{n_a-1}^T \end{bmatrix}.$$

We may conclude that  $P$  is  $(2n_a - 1) \times (4n_b - 2)$  and that

$$F = \overbrace{[P, P, \dots, P]}^{n_b \text{ blocks}} D_R^T.$$

Finally, substituting this expression for  $F$  into (5) yields

$$\tilde{T} = \underbrace{\text{diag}(W_{n_a} Z_{2n_a-1}^{n_a-1}, \dots, W_{n_a} Z_{2n_a-1}^0)}_{D_L} \begin{bmatrix} P & P & \dots & P \\ P & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ P & \dots & \dots & P \end{bmatrix} D_R^T. \quad (6)$$

In short, the matrix  $P$  is repeated within the larger matrix along  $n_a$  block rows and  $n_b$  block columns.

Recall from (4) that we wish to find the matrix  $\sum_{k=1}^s \tilde{a}_k \tilde{b}_k^T$  closest to  $\tilde{T}$  in the Frobenius norm, where  $\tilde{a}_k = \text{vec}(A_k)$ ,  $\tilde{b}_k = \text{vec}(B_k)$ , with the restriction that  $A_k$  is Toeplitz and  $B_k = B_k^{(t)} + B_k^{(h)}$  is Toeplitz-plus-Hankel.  $A_k$  is  $n_a \times n_a$  and is defined by its first row and column. Hence, we define the column vector

$$a_k = [\text{flip1r}(A_k(1,:))^T; A_k(2:n_a, 1)] \in \mathbb{R}^{2n_a-1}. \quad (7)$$

The matrix  $B_k^{(t)}$  is Toeplitz and  $n_b \times n_b$ , so is represented by its first row and column, too, and  $B_k^{(h)}$  is Hankel and so it is described by its first column and last row. Therefore, all entries of  $B_k$  can be determined from  $4n_b - 2$  numbers appearing in the vector

$$b_k = [\text{fliplr}(T_k^{(t)}(1, \cdot)^T; T_k^{(t)}(2:n_b, 1); T_k^{(h)}(\cdot, 1); T_k^{(h)}(n_b, 2:n_b)^T] \in \mathbb{R}^{4n_b-2}. \quad (8)$$

By a similar argument to the derivation of (6), it is readily shown that

$$\tilde{a}_k(\tilde{b}_k^{(t)} + \tilde{b}_k^{(h)})^T = D_L \left( \begin{bmatrix} a_j \\ a_j \\ \vdots \\ a_j \end{bmatrix} [b_k^T, b_k^T, \dots, b_k^T] \right) D_R^T. \quad (9)$$

Equations (6) and (9) give

$$\tilde{T} - \sum_{k=1}^s (\tilde{a}_k \tilde{b}_k^T) = D_L \left( \begin{bmatrix} P & P & \cdots & P \\ P & P & \cdots & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ P & \cdots & \cdots & P \end{bmatrix} - \sum_{k=1}^s \begin{bmatrix} a_k \\ a_k \\ \vdots \\ a_k \end{bmatrix} [b_k^T, b_k^T, \dots, b_k^T] \right) D_R^T. \quad (10)$$

Clearly the matrix in parentheses above, which we refer to as  $G$ , is not full rank. The following theorem makes this more concrete.

**Theorem 3.1** *There exist orthogonal  $Q_L, Q_R$  so that*

$$Q_L^T G Q_R = \sqrt{n_a n_b} \begin{bmatrix} P - \sum_{k=1}^s a_k b_k^T & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

*In particular, the orthogonal matrices have first block columns that appear as*

$$\sqrt{1/n} \begin{bmatrix} I_r \\ I_r \\ \vdots \\ I_r \end{bmatrix}$$

*with  $n = n_a$ ,  $r = 2n_a - 1$  for  $Q_L$  and  $n = n_b$ ,  $r = 4n_b - 2$  for  $Q_R$  where  $I_r$  denotes an  $r \times r$  identity matrix.*

*Proof:* The proof follows directly along the lines of the proof of Lemma 3.4 in [15] and therefore is omitted.  $\diamond$

Applying the theorem to (10), we have

$$\tilde{T} - \sum_{k=1}^s (\tilde{a}_k \tilde{b}_k^T) = D_L Q_L \left( \begin{bmatrix} P - \sum_{k=1}^s a_k b_k^T & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \right) Q_R^T D_R^T.$$

Recall that our goal is to minimize the expression on the left of this equation in the Frobenius norm. The Frobenius norm is invariant under orthogonal transformation. Suppose we have  $D_L Q_L = \tilde{Q}_L \tilde{R}_L$  and  $D_R Q_R = \tilde{Q}_R \tilde{R}_R$ . Then the Frobenius norm of the right-hand side above becomes

$$\left\| \tilde{R}_L \left( \begin{bmatrix} P - \sum_{k=1}^s a_k b_k^T & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \right) \tilde{R}_R \right\|_F = \|\tilde{R}_{L1} (P - \sum_{k=1}^s a_k b_k^T) \tilde{R}_{R1}\|_F, \quad (11)$$

where  $\tilde{R}_{L1}, \tilde{R}_{R1}$  are the leading  $(2n_b - 1) \times (2n_b - 1)$  and  $(4n_b - 2) \times (4n_b - 2)$  submatrices of  $\tilde{R}_L, \tilde{R}_R$ , respectively. Once  $\tilde{R}_{L1}, \tilde{R}_{R1}$  are known, the minimization problem can be solved with the help of the SVD of a relatively small matrix, as we explain in Section 4. Therefore, we derive a convenient form for these submatrices in the next subsection.

### 3.3 Forms of $\tilde{R}_{L1}, \tilde{R}_{R1}$

Theorem 3.1 gives us a formula for the first block-column of  $Q_L$ . Based on this formula and the definition of  $D_L$ , it is straightforward to show

$$\tilde{R}_{L1}^T \tilde{R}_{L1} = \frac{1}{n_a} \sum_{j=0}^{n_a-1} (Z_{2n_a-1}^j)^T W_{n_a}^T W_{n_a} Z_{2n_a-1}^j.$$

It is then easy to recover  $R_{L1}$  based on the following lemma.

**Lemma 3.1** For  $Z_{2n-1}$  the downshift matrix defined previously and  $W_n = [I_{n_a}, 0]$ ,

$$\sum_{j=0}^{n-1} (Z_{2n-1}^j)^T W_n^T W_n Z_{2n-1}^j = \text{diag}(1, 2, \dots, n, n-1, \dots, 1).$$

Proof: From the form of  $Z_{2n-1}$ , it can be verified that for  $j = 0, \dots, n$ ,

$$Z_{2n-1}^j = \left[ \overbrace{0, 0, \dots, 0}^j, \overbrace{e_1, e_2, \dots, e_{2n-1-j}}^{2n-1-j} \right].$$

Thus  $W_n Z_{2n-1}^j$  has the block form

$$W_n Z_{2n-1}^j = \left[ \underbrace{0}_j, I_n, \underbrace{0}_{n-1-j} \right].$$

This means that the  $j$ th term in the sum is a block diagonal matrix of the form

$$\begin{bmatrix} 0_j & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & 0_{n-1-j} \end{bmatrix},$$

where the subscripts denote the size of each square block. In particular, when  $j = 0$ , the identity block is the leading principal submatrix, and when  $j = n - 1$ , it is the trailing principal submatrix. Thus, a '1' appears in the  $k$ th or  $(k + n)$ th row for  $k = 1, \dots, n$ , exactly  $k$  times, and the lemma follows.  $\diamond$

Hence, from the lemma

$$\tilde{R}_{L1} = \frac{1}{\sqrt{n_a}} \text{diag}(\sqrt{1}, \sqrt{2}, \dots, \sqrt{n_a-1}, \sqrt{n_a}, \sqrt{n_a-1}, \dots, \sqrt{2}, \sqrt{1}). \quad (12)$$

Next we observe that the first block-column of  $D_R Q_R$  is

$$\frac{1}{\sqrt{n_b}} \begin{bmatrix} W_{n_b} Z_{2n_b-1}^{n_b-1} & W_{n_b} Z_{2n_b-1}^0 \\ W_{n_b} Z_{2n_b-1}^{n_b-2} & W_{n_b} Z_{2n_b-1}^1 \\ \vdots & \vdots \\ W_{n_b} Z_{2n_b-1}^0 & W_{n_b} Z_{2n_b-1}^{n_b-1} \end{bmatrix}.$$

Therefore,  $\tilde{R}_{R1}^T \tilde{R}_{R1}$  is the  $2 \times 2$  block matrix

$$\frac{1}{n_b} \begin{bmatrix} D & S \\ S^T & D \end{bmatrix},$$

of size  $(4n_b - 2) \times (4n_b - 2)$ . From Lemma 3.1, we have

$$D = \text{diag}(1, 2, \dots, n_b - 1, n_b, n_b - 1, \dots, 2, 1). \quad (13)$$

It is straightforward to show

$$S = \sum_{j=0}^{n_b-1} (Z_{2n_b-1}^j)^T W_{n_b}^T W_{n_b} Z_{2n_b-1}^{n_b-j-1}. \quad (14)$$

We now prove that the  $S$  block in the  $2 \times 2$  block matrix has a special form.

**Lemma 3.2** *Let  $Z_n, W_n$  be as before. Then*

$$\sum_{j=0}^{n-1} (Z_{2n-1}^j)^T W_n^T W_n Z_{2n-1}^{n-j-1}$$

*is equal to a Toeplitz-plus-Hankel matrix. For  $n$  odd, with*

*$v = \underbrace{[1, 0, 1, 0, \dots, 1, 0, \dots, 0]}_n$  and  $w = \underbrace{[1, 0, 1, \dots, 1, 0, \dots, 0]}_{n-2}$ , the Toeplitz-plus-Hankel matrix is*

$$\text{toeplitz}(v, v) - \text{hankel}(w, \text{fliplr}(w)).$$

*For  $n$  even, the matrix is given by the same command for  $v = \underbrace{[0, 1, 0, 1, 0, \dots, 1, 0, \dots, 0]}_n$  and  $w =$*

$$\underbrace{[0, 1, 0, 1, 0, \dots, 1, 0, \dots, 0]}_{n-2}.$$

*Proof:* We are given the form of  $W_n Z_{2n-1}^j$  within the proof of Lemma 3.1. Similarly, for  $j = 0, \dots, n-1$ ,

$$W_n Z_{2n-1}^{n-j-1} = \left[ \underbrace{0}_{n-j-1}, I_n, \underbrace{0}_j \right].$$

It follows that for  $j = 0, \dots, n$ ,  $(W_n Z_{2n-1}^j)^T (W_n Z_{2n-1}^{n-j-1})$  will be a  $(2n-1) \times (2n-1)$  matrix with all zero entries except for an embedded  $n \times n$  identity matrix, where the (1,1) entry of the identity matrix is located at row  $j+1$ , column  $n-j$ . As  $j$  increases, the identity matrix moves down one unit and to the right one unit. This means that  $(W_n Z_{2n-1}^j)^T (W_n Z_{2n-1}^{n-j-1})$  has one non-zero diagonal, the  $(n-2j-1)$ st diagonal, and for no  $j$  do these matrices overlap. The result of summing over  $j$  is a diamond-shaped pattern of 1's and 0's, with the point of the diamond at the first row, column  $n$ . It is easily established that the Toeplitz-plus-Hankel matrix in the statement of the lemma yields the same matrix.  $\diamond$

Using the lemma with  $n = n_b$ , this means that the matrix  $S$  is a symmetric, Toeplitz-plus-Hankel matrix. Next, we look for a triangular factorization of the  $2 \times 2$  block matrix.

**Lemma 3.3** *The  $2 \times 2$  block matrix  $\tilde{R}_{R1}^T \tilde{R}_{R1}$  can be factored as*

$$\frac{1}{n_b} \begin{bmatrix} D^{1/2} & 0 \\ SD^{-1/2} & C^T \end{bmatrix} \begin{bmatrix} D^{1/2} & D^{-1/2}S \\ 0 & C \end{bmatrix},$$

*where  $C$  is an upper triangular matrix of rank  $2n_b - 3$  such that  $C^T C = D - SD^{-1}S$  and the last 2 rows of  $C$  are 0. In particular,  $C$  can be obtained by applying the Cholesky factorization algorithm until the first  $2n_b - 3$  non-zero columns of  $C$  are obtained.*

*Proof:* It is readily checked that the proposed block factorization matches  $\tilde{R}_{R1}^T \tilde{R}_{R1}$  in all four blocks, provided that we can show such a matrix  $C$  exists. To show that  $C$  exists, first consider the matrix  $D^{-1}S$ . From the forms of  $D$  and  $S$ , it is easy to see that row  $i$  for  $i = 1, \dots, n$  of  $D^{-1}S$ , which we will denote by  $r_i$ , is the  $i$ th row of  $S$ , which has exactly  $i$  1's appearing in its row, multiplied by  $1/i$ . Row  $i+n$ , for  $i = 1, \dots, n-1$ , of  $D^{-1}S$  is the same as row  $i$ . It follows that the sum of the entries in row  $i$  of  $D^{-1}S$  is 1.

Let  $\hat{r}_i$  denote the  $i$ th row of  $SD^{-1}S$ . Then it is straightforward to show

$$\hat{r}_i = \begin{cases} r_n + 2 \sum_{j=1}^{\frac{i-1}{2}} r_{n-2j} & \text{if } i \text{ is odd} \\ 2 \sum_{j=0}^{\frac{i}{2}} r_{n-(2j+1)} & \text{if } i \text{ is even} \end{cases}$$

Since the sum of the entries in each vector  $r_k$  is 1, it follows by the formula that the sum of the entries in each vector  $\hat{r}_i$  is  $i$ . It also follows that the non-zero entries in  $SD^{-1}S$  occur in a ‘‘checkerboard’’ pattern, with odd rows having entries only in odd numbered columns and even rows having entries only in even numbered columns. Clearly, the entries of  $SD^{-1}S$  are all positive. An example for  $n_b = 4$  is illustrated in Figure 1.

Since  $D$  is a diagonal matrix with  $(i, i)$  entry  $i$ , the sum of the entries in row  $i$  of  $D - SD^{-1}S$  is 0. In particular,  $D - SD^{-1}S$  has a non-trivial null space of dimension 2 which is spanned by the vectors  $[1, 0, 1, 0, 1, \dots, 1]^T$  and  $[0, 1, 0, \dots, 1, 0]^T$ . This says that the two right-most columns of the matrix are linear combinations of the preceding columns. Furthermore, Gershgorin's theorem can be used to show that all the eigenvalues are non-negative and in particular, that the leading  $2n_b - 3$  principal submatrices are positive definite.

$$\begin{bmatrix} 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\ 0 & 2/3 & 0 & 2/3 & 0 & 2/3 & 0 \\ 1/4 & 0 & 5/4 & 0 & 5/4 & 0 & 1/4 \\ 0 & 2/3 & 0 & 8/3 & 0 & 2/3 & 0 \\ 1/4 & 0 & 5/4 & 0 & 5/4 & 0 & 1/4 \\ 0 & 2/3 & 0 & 2/3 & 0 & 2/3 & 0 \\ 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \end{bmatrix}$$

Figure 1: The matrix  $SD^{-1}S$  for  $n_b = 4$ .

We want to factor  $D - SD^{-1}S$  into  $C^T C$ , with  $C$  upper triangular, even though the matrix is rank deficient. Because of the checkerboard pattern, odd-numbered steps in a Cholesky factorization (up to step  $2n_b - 3$ ) of  $D - SD^{-1}S$  would be independent from even-numbered steps. That is, the contributions to the triangular factor from the  $n \times n$  submatrix of  $D - SD^{-1}S$  (which we call  $G_1$ ) obtained by selecting only odd-numbered rows and columns is independent from the contributions due to the  $(n - 1) \times (n - 1)$  submatrix (referred to as  $G_2$ ) comprised of the even numbered rows and columns. Thus, it is sufficient to find triangular factors of  $G_1$  and  $G_2$ , and interleave the results.

By the argument above, all the leading principal submatrices of  $G_1$  up to order  $n - 1$  are positive definite,  $G_1$  has rank  $n - 1$  and the right-most column is obtained by summing all the previous  $n - 1$  columns for  $G_1$  and negating. Similarly, all the leading principal submatrices of  $G_2$  up to order  $n - 2$  are positive definite,  $G_2$  has rank  $n - 2$  and the right-most column is obtained by summing all the previous submatrices of  $G_2$  up to  $n - 2$  and negating. It follows that the Cholesky factorizations of  $G_1$  and  $G_2$  can be computed up to steps  $n - 1$  and  $n - 2$ , respectively, and that the last columns of the factor will, after those steps, be zero. Thus, the matrix  $D - SD^{-1}S$  can be written as  $C^T C$  for an upper triangular matrix  $C$ , where the first  $2n_b - 3$  columns are obtained by applying  $2n_b - 3$  steps of the Cholesky factorization algorithm to  $D - SD^{-1}S$  and the last two columns are zero.  $\diamond$

The lemma then gives us a formula for computing  $\tilde{R}_{R1}$ :

$$\tilde{R}_{R1} = \frac{1}{\sqrt{n_b}} \begin{bmatrix} D^{1/2} & D^{-1/2}S \\ 0 & C \end{bmatrix} \quad (15)$$

where  $C$  is as defined in Lemma 3.3 and  $D$  and  $S$  are defined in (13), (14), respectively.

## 4 The Algorithm

Recall from (11) that

$$\begin{aligned} \|\tilde{T} - \sum_{k=1}^s (\tilde{a}_k \tilde{b}_k^T)\|_F &= \|\tilde{R}_{L1} P \tilde{R}_{R1}^T - \sum_{k=1}^s (\tilde{R}_{L1} a_k)(b_k^T \tilde{R}_{R1}^T)\|_F \\ &\equiv \|\bar{P} - \sum_{k=1}^s \bar{a}_k \bar{b}_k^T\|_F. \end{aligned}$$

It is well known that the  $\bar{a}_k, \bar{b}_k$  that minimize the Frobenius norm of the previous expression can be determined from the SVD of  $\bar{P}$  [5]. Hence, our algorithm is as follows.

### Algorithm 1

1. Compute  $\tilde{R}_{L1}, \tilde{R}_{R1}$  according to (12) and (15).
2. Form  $\bar{P} = \tilde{R}_{L1} P \tilde{R}_{R1}^T$
3. Find  $s$  terms in the SVD for  $\bar{P}$ , and thus the best rank- $s$  approximation, in the Frobenius norm:

$$\bar{P} \approx \sum_{k=1}^s \sigma_k u_k v_k^T.$$

4. Set  $\bar{a}_k = \sqrt{\sigma_k} u_k, \bar{b}_k = \sqrt{\sigma_k} v_k$
5. Solve  $\tilde{R}_{L1} a_k = \bar{a}_k, \tilde{R}_{R1} b_k = \bar{b}_k$  by substitution.
6. Determine the rows and columns of  $A_k, B_k^{(t)}, B_k^{(h)}$  from (7) and (8) as appropriate.

## 4.1 Special Case: dense BTTB matrices

Algorithm 1, which approximates a BTTHB matrix by a sum of Kronecker products, can be easily adapted to handle the special case when the blocks that comprise the block-Toeplitz matrix are only Toeplitz, rather than Toeplitz-plus-Hankel. The adjustments that need to be made for the case when the Hankel components of each block are zero are as follows.

- $P$  should have only  $2n_a - 1$  columns.
- $b_k$  should contain only the first half of the terms in (8).
- $D_R$  becomes the block diagonal matrix

$$D_R = \text{diag}[(Z_{2n_b-1}^{n_b-m-1})^T W_{n_b}^T, \dots, (Z_{2n_b-1}^0)^T W_{n_b}^T].$$

- The definition of  $r$  for  $Q_R$  in Theorem 3.1 becomes  $2n_b - 1$ .

These facts imply that in the special case,

$$\tilde{R}_{R1} = \frac{1}{\sqrt{n_b}} D^{1/2}, \quad (16)$$

where  $D$  is from (13). Accordingly, the only changes that need to be made to Algorithm 1 to accommodate the special case is to change the reference in Step 1 to equation (16) and to realize that  $b_k$  contains only the information needed to build  $B_k^{(t)}$ .

## 4.2 General Case: dense BTHTHB matrices

Algorithm 1 can be generalized for the case of BTHTHB matrices. The notation becomes very messy when trying to describe the algorithm in detail, so in this subsection we only provide some comments on how our algorithm can be extended, and leave the details to the interested reader.

First, a BTHTHB matrix can be written as

$$T + H = \begin{bmatrix} T_0 & T_{-1} & \cdots & T_{1-n_a} \\ T_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & T_{-1} \\ T_{n_a-1} & \cdots & T_1 & T_0 \end{bmatrix} + \begin{bmatrix} H_0 & \cdots & H_{n_a-2} & H_{n_a-1} \\ \vdots & \ddots & \ddots & H_{n_a} \\ H_{n_a-2} & \ddots & \ddots & \vdots \\ H_{n_a-1} & H_{n_a} & \cdots & H_{2n_a-2} \end{bmatrix}$$

where each  $T_k$  and  $H_k$  is a Toeplitz-plus-Hankel matrix. Thus, we write

$$T_k = T_k^{(t)} + T_k^{(h)} \quad \text{and} \quad H_k = H_k^{(t)} + H_k^{(h)}.$$

The approximation we seek in this case has the form:

$$\sum_k (A_k^{(t)} + A_k^{(h)}) \otimes (B_k^{(t)} + B_k^{(h)})$$

where  $A_k^{(t)}$  and  $B_k^{(t)}$  are Toeplitz matrices, and  $A_k^{(h)}$  and  $B_k^{(h)}$  are Hankel matrices.

The key observation needed to generalize Algorithm 1 is that the array  $P$  is now given by

$$P = \begin{bmatrix} p_{1-n_a}^T \\ \vdots \\ p_{n_a-1}^T \\ q_0^T \\ \vdots \\ q_{2n_a-2} \end{bmatrix} \in \mathfrak{R}^{(4n_a-2) \times (4n_b-2)}$$

where  $p_k$  are defined from  $T_k$ , as described in section 3.2, and  $q_k$  are similarly defined from  $H_k$ . Specifically,  $q_k^T = [\hat{x}_k^T, \hat{y}_k^T]$ , where  $\hat{x}_k$  is a  $2n_b - 1$  length vector that contains the entries in the first row of  $H_k^{(t)}$ , in reverse order, followed by the remainder of the terms in the first column, and  $\hat{y}_k$  is a  $2n_b - 1$  length vector containing the entries in the first column of  $H_k^{(h)}$  followed by the remaining entries in the last row.

Once  $P$  is defined, other modifications to the algorithm are fairly obvious. For example, the matrix  $\tilde{R}_{L1}$  should be computed using the same approach as used for  $\tilde{R}_{R1}$ . Additionally, the columns for  $B_k^{(t)}$  and  $B_k^{(h)}$  are constructed as before, using (8), and a similar approach is used for  $A_k^{(t)}$  and  $A_k^{(h)}$ .

## 5 Numerical Results

Our algorithm computes the best approximation of a given matrix as a sum of  $s$  Kronecker products. As an application, we consider the use of these sums to derive preconditioners for the linear system (1) for which we desire to find a solution and for (2) for which we desire only a regularized solution. As we note in the discussion below, the derivation of the preconditioner for the linear system case is slightly different than for the case when we want only a regularized solution.

For simplicity, we will focus the discussion on left preconditioning and assume that we apply an iterative solver such as the conjugate gradient method. We use the MRH algorithm [6] for the example from diffuse optical tomography where the matrix is symmetric, and we use the CGLS algorithm [1] for the atmospheric imaging example when the matrix is not symmetric. We begin with a brief discussion of preconditioning, followed by our specific numerical examples.

### 5.1 Preconditioning basics

A system  $M^{-1}Kf = M^{-1}g$  is said to be left preconditioned by an invertible matrix  $M$ . Note that this system has the same solution as the unpreconditioned system  $Kf = g$ .

If there is *no noise* in the right-hand side and if the goal is to approximate  $f$  with as few iterations as possible (e.g. (1)), then we desire  $M \approx K$  so that the singular value or eigenvalue spectrum of  $M^{-1}K$  is clustered around a point bounded away from zero. We need to be able to compute  $M^{-1}$  explicitly or we need to be able to perform solves with  $M$  quickly in order for  $M$  to be considered a useful preconditioner.

We can obtain such a preconditioner in the following way. We use Algorithm 1 to construct an SVD approximation of  $K$ . For example, if we choose  $s = 1$  in our approximation algorithm, then properties of Kronecker products imply

$$A_1 \otimes B_1 = U_{a_1} \Sigma_{a_1} V_{a_1}^T \otimes U_{b_1} \Sigma_{b_1} V_{b_1}^T = (U_{a_1} \otimes U_{b_1})(\Sigma_{a_1} \otimes \Sigma_{b_1})(V_{a_1}^T \otimes V_{b_1}^T).$$

Therefore, we have

$$K \approx (U_{a_1} \otimes U_{b_1})(\Sigma_{a_1} \otimes \Sigma_{b_1})(V_{a_1}^T \otimes V_{b_1}^T) = U \Sigma V^T$$

so if we set  $M = U \Sigma V^T$ , we have  $M \approx K$ , the spectrum of  $M^{-1}K$  should be clustered around 1, and  $M^{-1}$  can be applied efficiently using the properties of Kronecker products.

However, if  $A_1 \otimes B_1$  is not a very good approximation of  $K$ , then it may be necessary to choose  $s > 1$  in Algorithm 1. Unfortunately  $s > 1$  presents some computational difficulties since, generally, the inverse of the sum of Kronecker products is not the sum of the inverses, nor is it usually possible to compute the SVD of the sum of Kronecker products directly from the SVDs of each term in the sum. In this case, we could use an approach advocated in [11, 15]: compute the SVD of the first term in the sum (this corresponds to taking  $s = 1$  in Algorithm 1) as  $A_1 = U_{a_1} \Sigma_{a_1} V_{a_1}^T$ ,  $B_1 = V_{b_1} \Sigma_{b_1} V_{b_1}^T$ . Then an approximate SVD  $U \Sigma V^T$  of the sum  $\hat{K} = \sum_{i=1}^s A_i \otimes B_i$  is given by  $U = U_{a_1} \otimes U_{b_1}$ ,  $V = V_{a_1} \otimes V_{b_1}$ ,  $\Sigma = \text{diag}(U^T \hat{K} V)$ . The fact that  $\hat{K}$  is a sum of Kronecker products makes it relatively inexpensive to compute  $\Sigma$ . We then design our preconditioner based on the second level approximation to  $K$  by  $(U_{a_1} \otimes U_{b_1}) \Sigma (V_{a_1}^T \otimes V_{b_1}^T)$ .

On the other hand, if  $K$  is an ill-conditioned matrix arising from discretization of an *ill-posed problem*, then we must consider the model given by (2), and incorporate a regularization technique. There are many choices for regularizing the problem [4, 8, 20]; in this paper we use iterative regularization. That is, we apply a standard iterative method, such as MRH or CGLS, to the linear system  $Kf + e = g$ , and terminate the iteration when a good approximation of the solution has been computed. In this case the preconditioned system has the form

$$M^{-1}Kf + M^{-1}e = M^{-1}g.$$

In order to ensure  $M^{-1}e$  does not dominate and contaminate the solution, we only want  $M$  to approximate  $K$  in such a way that the largest elements of the spectrum of  $M^{-1}K$  are clustered around 1 while the small elements of the spectrum are the same as those of  $K$ . This can be done by truncating the approximate singular value decomposition we derived above. Specifically, if  $K \approx U \Sigma V^T$ , then we use the preconditioner

$$M = U \Sigma_\tau V^T,$$

where the diagonal elements of  $\Sigma_\tau$  are

$$[\Sigma_\tau]_{ii} = \begin{cases} [\Sigma]_{ii} & [\Sigma]_{ii} \geq \tau \\ 1 & \text{otherwise,} \end{cases}$$

where  $0 \leq \tau \leq [\Sigma]_{11}$ . This approach of using a rank-revealing approximate factorization then truncating has been used effectively for circulant approximations of  $K$ ; for details, see [7, 13].

In summary, the preconditioner is derived as follows:

1. Use Algorithm 1 to determine  $K \approx \sum_{i=1}^s A_i \otimes B_i$ .
2. Compute an approximate (exact, if  $s = 1$ ) SVD for the sum as

$$(U_{a_1} \otimes U_{b_1})\Sigma(V_{a_1}^T \otimes V_{b_1}^T) = U\Sigma V^T$$

with  $\Sigma$  and other quantities as defined above.

3. Set  $M = U\Sigma_\tau V^T$ , where  $\tau = 0$  for a problem that does not need regularization, and  $\tau > 0$  is determined by a regularization parameter selection method for problems that need regularizing.

We use this preconditioning approach in the following examples to illustrate the effectiveness of our approximation algorithm. The experiments in this section were computed using MATLAB in IEEE double precision arithmetic, and the iterative image restoration codes provided by [16].

## 5.2 Example from 2D Diffuse Optical Tomography

In diffuse optical tomographic imaging, the goal is to reconstruct the optical properties of tissue (e.g. optical absorption) from diffuse optical data taken at the surface of the region of interest. When a nonlinear, integral equation formulation is used to model the relation of the input to the output, then in the reconstruction phase, one must solve a large linear system of equations at each step of the nonlinear reconstruction algorithm [12]. In this example, we consider the development of preconditioners for such systems.

We assume that the region to be imaged is modeled as an infinite strip of width  $d$  and we use an extrapolated boundary condition (which is essentially a method of images approach about an imaginary line distance  $z_b$  away from the boundary). In this case, the matrix is derived as follows. Let  $s, t$  be defined as  $s = (x_1, z_1)$  and  $t = (x_2, z_2)$ . (Note,  $s, t, x_1, x_2, z_1, z_2$  used in this section do not refer to any previously defined notation.) Let

$$\rho_j = \sqrt{(x_2 - x_1)^2 + (z_2 - z_1 + j(2d + 4z_b))^2}$$

and

$$\hat{\rho}_j = \sqrt{(x_2 - x_1)^2 + (z_2 + z_1 + j(2d + 4z_b) + 2z_b)^2}.$$

Then

$$\hat{k}(s, t) = c \left( \gamma - \exp(-\sigma \hat{\rho}_j) / \hat{\rho}_j + \sum_{j=-J, j \neq 0}^J [\exp(-\sigma \rho_j) / (4\pi \rho_j) - \exp(-\sigma \hat{\rho}_j) / (4\pi \hat{\rho}_j)] \right),$$

where

$$\gamma = \begin{cases} (3\mu_{sp})^2(1/\sigma^2)(1 - (1 - \sigma R) \exp(\sigma R)) & s = t \\ (3\mu_{sp}) \exp(\sigma \rho_0) / (4\pi \rho_0) & s \neq t \end{cases}$$

Here,  $c$  is the volume of the voxel,  $\mu_{sp}$  is the reduced scattering coefficient,  $\sigma$  is the imaginary part of the wave number and  $R = (\frac{c}{4/3\pi})^{1/3}$ .

The matrix  $\hat{K}$  is defined by taking samples of  $\hat{k}(s, t)$  for  $s$  and  $t$  on the same rectangular grid and ordering the values according to a lexicographical ordering in both variables. In other words, the rows of  $\hat{K}$  correspond to varying  $s$  across the grid and the columns to varying  $t$ . Clearly,  $\hat{K}$  is a BTTHB matrix if the ordering is done first in the  $z$  direction, then in the  $x$  direction.

However, in the DOT application, the system we need to solve [12] is

$$\tilde{K}f = g, \quad \tilde{K} = I - \hat{K}\mathcal{D},$$

where  $\mathcal{D}$  is a diagonal matrix containing the discretized version of a vector of the optical absorption sampled at the same points on the grid.  $\tilde{K}$  as written is not symmetric, but the system can be re-written to form the symmetrized system:

$$K\hat{f} = (\mathcal{D}^{-1} - \hat{K})\hat{f} = g, \quad \hat{f} = \mathcal{D}f.$$

We assume that absorption is piecewise constant, which leads to  $\mathcal{D} = \mu_1 I + \mu_2 \Gamma$ , where  $\Gamma$  is a relatively low-rank diagonal matrix and  $\mu_1, \mu_2$  are constants. Therefore,  $K \approx \frac{1}{\mu_1} I - \hat{K} = \bar{K}$ , where the latter is

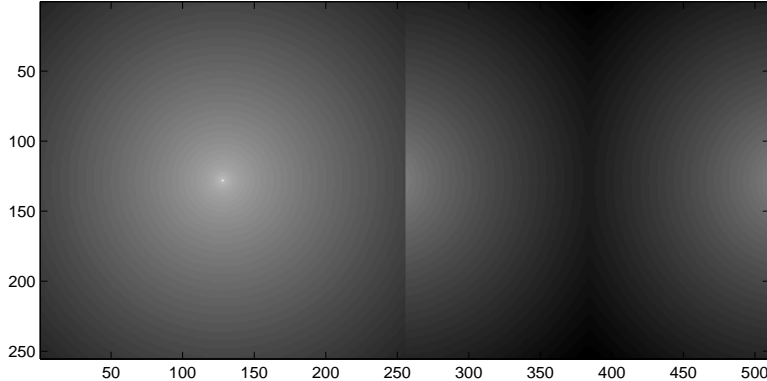


Figure 2: The matrix  $P$  needed to construct the Kronecker product approximation of  $\bar{K}$  (see section 3.2) for Example 1. The  $255 \times 255$  submatrix on the left corresponds to the Toeplitz component of  $K$ , while the  $255 \times 255$  submatrix on the right corresponds to the Hankel component.

clearly a BTTHB matrix. We therefore feed  $\bar{K}$  into the algorithm that determines a preconditioner in hopes that  $M \approx \bar{K} \approx K$ . The matrix  $P$  needed to construct the Kronecker product approximation of  $\bar{K}$  (see section 3.2) is shown in Figure 2. Since the matrix and preconditioner are symmetric, we use the iterative method MR-II and its preconditioned analogue [6], which requires only one matrix-vector product (and preconditioner application) per iteration.

It is important to note that new linear systems must be solved for each outer iteration of the nonlinear imaging algorithm, but as  $\mu_1$  does not change much, the preconditioner can be formed once and reused. Furthermore, due to noise in the data underlying the nonlinear problem, the linear systems need not be solved terribly accurately. We therefore stop iterating according to  $\|K\hat{f}_j - g\|/\|g\| < 5 * 10^{-6}$ .

In this problem, we set  $\mu_1 = .008, \mu_2 = .08, \sigma = 1.2247, J = 2$ . The discrete values of  $x$  used were  $[-3 : .05 : 3]$  and for  $z$  they were  $[.2 : .05 : 6.55]$ , where the units are in cm.  $K$ , therefore, is  $128^2 \times 128^2$ . The strip thickness was 7.55cm, and the value of  $z_b$  was  $1.84 \times 10^{-1}$  cm. The true solution “image” was a  $128 \times 128$  array of zeros, except for 1’s in positions rows 72 to 80, columns 60 to 63. We then vectorized this image to obtain  $f$  and computed  $b = K\mathcal{D}f$ .

In Figure 3, we display the largest 16 values  $\sigma_s$  from Algorithm 1. (Note that these singular values are *not* the same as the singular values that are used to define  $M$ .) Although there is a clear gap between the first and second terms, using a preconditioner defined by taking  $s = 1$  is not helpful in terms of speeding the convergence, as shown in Figure 4. In fact, in Figure 4, we see that increasing  $s$  up to a value of 7 has a beneficial effect on the convergence rate. However, increasing  $s$  beyond 7 has no additional effect on the convergence rate. It appears that with preconditioning using  $s = 7$ , we achieve convergence in 3 iterations, whereas it takes 12 iterations with no preconditioning. It should be noted that although there is a cost associated with applying the preconditioner, it is less than the cost of a matrix-vector product with  $K$ . Furthermore, in the application being considered, not only is  $\mu_1$  virtually unchanged at each step of the nonlinear solver, but at each step of the nonlinear solver, several systems involving the same  $K$  but different right-hand sides must be solved. Therefore the cost of setting up the preconditioner is negligible, and the savings provided by the preconditioning is significant over many steps.

### 5.3 Example from Image Deblurring

In this example we consider the problem of obtaining an image of an astronomical object from a ground based telescope. In such problems the observed image is degraded by blurring (caused by atmospheric turbulence and optical limitations of the telescope mirrors) and noise. The image formation processes is modeled by (2), where  $g$  represents the observed (degraded) image,  $f$  represents the true image, and  $e$  is noise that enters during the data collection process.  $K$  is a structured matrix that describes the blurring operation, and is often given implicitly in terms of a *point spread function* (PSF). A PSF is an image (i.e., matrix of pixel values) of a point source object, such as a single bright star, and provides the essential information to construct  $K$ . In many cases the PSF has a (small) finite nonzero support, which means the matrix  $K$  has the banded structure described in [11, 15]. However, there are situations in which the PSF extends outside the boundaries of the field of view, leading to a dense matrix  $K$ . Figure

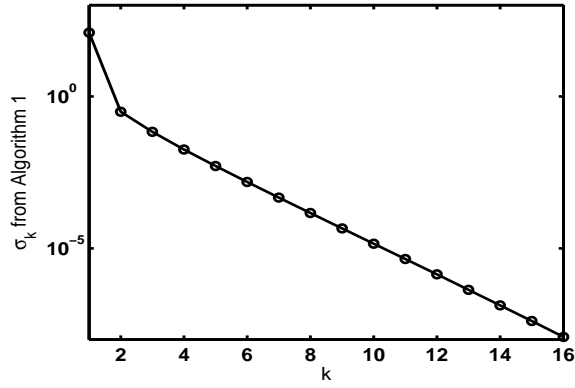


Figure 3: The values  $\sigma_k$  from Algorithm 1, example 1. Note these are the singular values for  $\bar{P}$ , not for the matrix approximation to  $K$ .

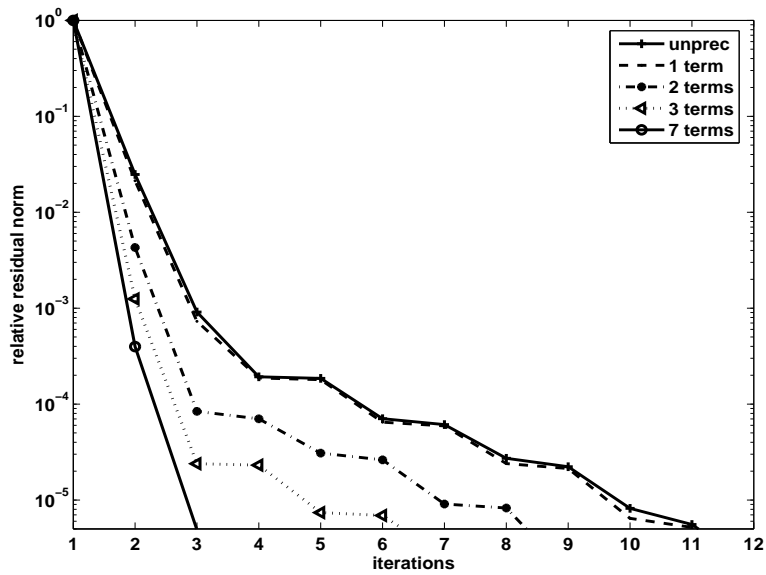


Figure 4: Convergence history  $\|K\hat{f}_j - g\|_2/\|g\|_2$  for unpreconditioned MR-II vs. preconditioned MR-II using preconditioners defined using different values of  $s$ .

5 shows examples of two PSFs. Note that even in the non-banded case shown on the right, the values of the PSF become small away from its center. Thus we might be tempted to crop these entries and form a banded approximation of  $K$ . As we see below, this may lead to poor reconstructions.

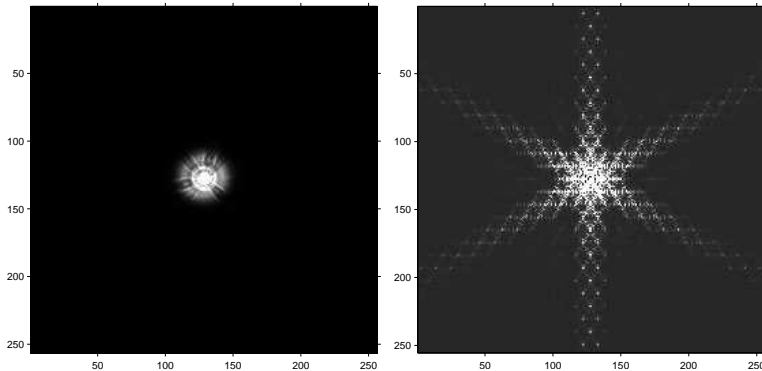


Figure 5: Two example PSFs. The left shows a PSF with finite support, leading to a banded matrix  $K$ . The right shows a PSF with support extending to the boundaries of the image, leading to a dense matrix  $K$ .

In the numerical results reported here, we use the image shown on the right in Figure 5 which is an example of a Keck telescope PSF [14]. To generate a test problem, we begin with the simulated star cluster image shown on the left of Figure 6, which was obtained from the *Space Telescope Science Institute*; this is our true image,  $f$ . We then use the  $255 \times 255$  Keck PSF shown on the right of Figure 5 for the array  $P$  that defines a dense  $128^2 \times 128^2$  BTTB matrix  $K$  (actually, we do not construct the matrix explicitly, but rather use  $P$  and FFTs to efficiently perform matrix-vector multiplications with  $K$ ). Given the matrix  $K$ , we construct  $g$  by computing  $g = Kf + e$ , where the entries of  $e$  are random values, normally distributed with mean 0 and variance 1, and scaled so that

$$\frac{\|e\|_2}{\|Kf\|_2} = 0.001.$$

The resulting blurred, noisy image is shown on the right of Figure 6.

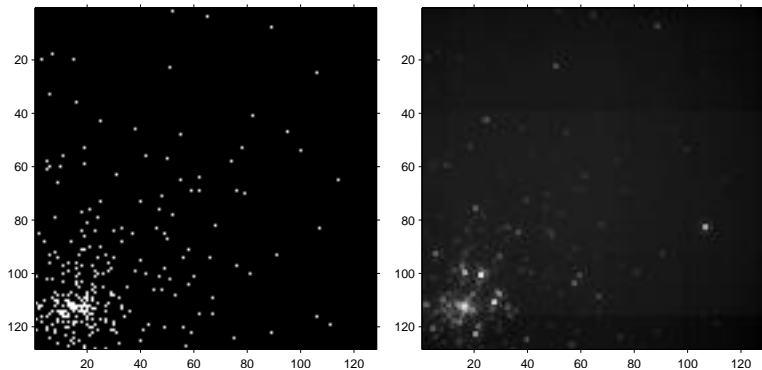


Figure 6: True image scene,  $f$ , on the left, and the blurred noisy image,  $g$ , on the right.

With this given data, we use the iterative method PCGLS to compute regularized approximations of  $f$ . We construct a regularized preconditioner as described above, using  $s = 1$  in Algorithm 1, and a truncation tolerance  $\tau = 0.001$ . To illustrate the effectiveness of this preconditioner, we compare the convergence behavior when no preconditioning is used, and when using the regularized circulant preconditioner (again, with  $\tau = 0.001$ ) described in [7]. We also consider cropping the small entries of  $P$  to obtain a reduced size,  $128 \times 128$  PSF, and obtain a banded approximation of  $K$ .

Figure 7 shows the convergence history of each approach; that is, we show plots of the relative errors

$$\frac{\|f - f_k\|_2}{\|f\|_2}$$

at each iteration, where  $f$  is the true solution, and  $f_k$  is the computed solution at the  $k$ th iteration. Note that due to the noise in the data, the relative errors will not approach zero, but rather approach a minimum value and then diverge as noise gets mixed into the iterates. Because we are interested in the convergence history of the early iterations, we show only the first 200 iterations. However, in Table 1 we show the number of iterations needed to obtain a minimum relative error reconstruction. The computed reconstructions are shown, along with the true solution, in Figure 8. Because of space considerations, and because it is an obviously poor approximation of the true image, we do not show the computed reconstruction when using the banded approximation of  $K$ .

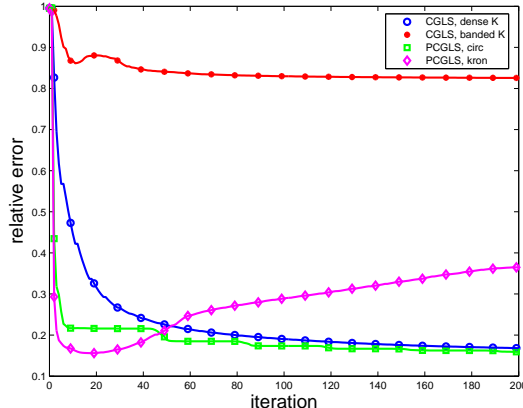


Figure 7: Convergence history of the first 200 iterations of CGLS vs. PCGLS.

|                      | CG, dense $K$ | CG, banded $K$ | PCG, circ | PCG, kron |
|----------------------|---------------|----------------|-----------|-----------|
| min relative error   | 0.1540        | 0.8218         | 0.1541    | 0.1561    |
| number of iterations | 578           | 1000           | 393       | 18        |

Table 1: Minimum relative errors attained by each approach, and the number of iterations needed to achieve this result.

These results clearly show the importance of using the dense matrix  $K$ , and not its banded approximation. Moreover, we see that the preconditioner constructed from our Kronecker product approximation algorithm is very effective in reducing the number of iterations needed to compute a good reconstruction.

## 6 Conclusions

We have introduced an algorithm for approximating dense BTTHB matrices by sums of Kronecker products, and we have shown this approximation to be optimal in the Frobenius norm when there is only one term in the sum. Furthermore, we showed how this approximation could be used to construct preconditioners for linear systems and discrete ill-posed problems involving matrices with this structure. We showed that it is straightforward to generalize our algorithm to adapt the approximation procedure for BTTB and BTHTHB matrices. Our numerical experiments illustrated the effectiveness of the approximations as preconditioners in two important image processing applications. In future work, we will consider Kronecker approximations for dense, structured matrices in three-dimensional imaging applications.

## References

- [1] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA, 1996.
- [2] D. Boas, D. Brooks, R. Gaudette, T. Gaudette, E. Miller, and Q. Zhang. Photon migration imaging (pmi) toolbox. freely available at <http://www.nmr.mgh.harvard.edu/DOT/resources/toolbox.htm>.

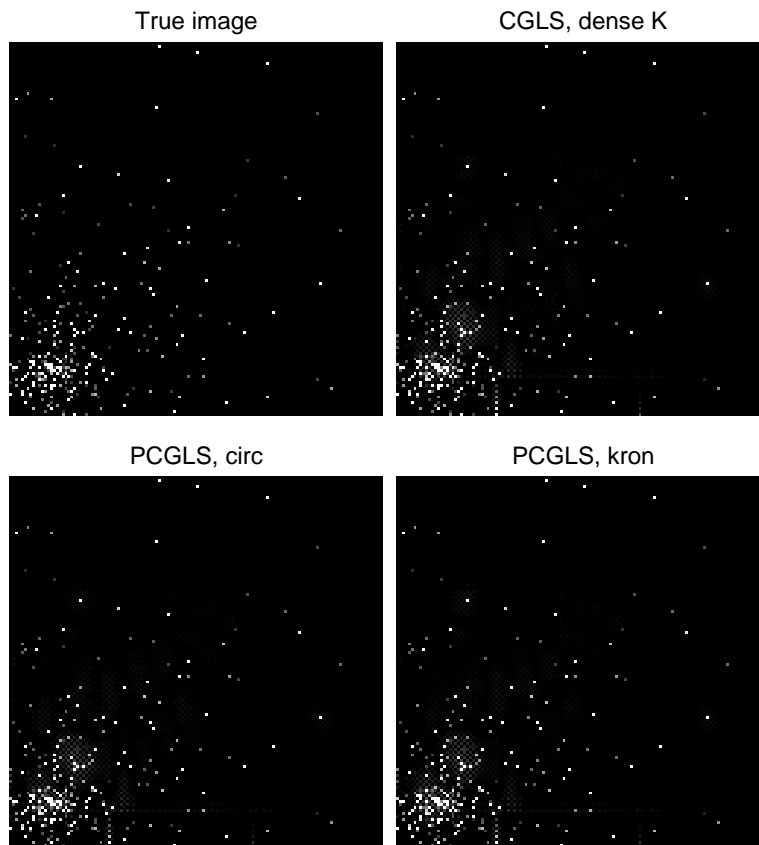


Figure 8: Computed reconstructions for each method. Upper left is the true solution, upper right is the CGLS reconstruction at iteration 578, lower left is the PCGLS with circulant preconditioner reconstruction at iteration 393, and lower right is the PCGLS with Kronecker product preconditioner reconstruction at iteration 18.

- [3] A. Diaspro, M. Corosu, P. Ramoino, and M. Robello. Two-photon excitation imaging based on a compact scanning head. *IEEE Engineering in Medicine and Biology*, pages 18–22, September/October 1999.
- [4] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, Dordrecht, 2000.
- [5] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.
- [6] M. Hanke and J. G. Nagy. Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques. *Inverse Problems*, 12:157–173, 1996.
- [7] M. Hanke, J. G. Nagy, and R. J. Plemmons. Preconditioned iterative regularization. In L. Reichel, A. Ruttan, and R. S. Varga, editors, *Numerical Linear Algebra*, pages 141–163. de Gruyter, Berlin, 1993.
- [8] P. C. Hansen. *Rank-deficient and discrete ill-posed problems*. SIAM, Philadelphia, PA, 1997.
- [9] J. W. Hardy. Adaptive optics. *Scientific American*, 270(6):60–65, 1994.
- [10] R. Haskell, L. Svasand, T. Tsay, T. Feng, M. McAdams, and B. Tromberg. Boundary conditions for the diffusion equation in radiative transfer. *J. Opt. Soc. Am. A.*, 11:2727–2741, 1994.
- [11] J. Kamm and J. G. Nagy. Optimal Kronecker product approximation of block Toeplitz matrices. *SIAM J. Matrix Anal. Appl.*, 22:155–172, 2000.
- [12] M. Kilmer, E. Miller, A. Barbaro, and D. Boas. Three-dimensional shape-based imaging of absorption perturbation for diffuse optical tomography. *Applied Optics*, 42(16), 2003.
- [13] Misha E. Kilmer. Cauchy-like preconditioners for 2-dimensional ill-posed problems. *SIAM J. Matrix Anal. Appl.*, 20(3), 1999.
- [14] F. Marchis. Theoretical PSFs for the KECK telescope and AO instruments. Center for Adaptive Optics (CfAO), University of California, Berkeley, <http://astron.berkeley.edu/~fmarchis/Science/Keck/PerfectPSF>, 2001.
- [15] J. G. Nagy, M. K. Ng, and L. C. Perrone. Kronecker product approximation for image restoration with reflexive boundary conditions. *SIAM J. Matrix Anal. Appl.*, 25:829–841, 2004.
- [16] J. G. Nagy, K. M. Palmer, and L. Perrone. Iterative methods for image deblurring: A Matlab object oriented approach. *Numerical Algorithms*, 36:73–93, 2004.
- [17] F. Natterer. *The Mathematics of Computerized Tomography*. SIAM, Philadelphia, PA, 2001.
- [18] M. Ng, R. H. Chan, and W. Tang. A fast algorithm for deblurring models with neumann boundary conditions. *SIAM J. Sci. Comput.*, 21:851–866, 1999.
- [19] C. F. Van Loan and N. P. Pitsianis. Approximation with Kronecker products. In M. S. Moonen and G. H. Golub, editors, *Linear Algebra for Large Scale and Real Time Applications*, pages 293–314. Kluwer Publications, 1993.
- [20] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, PA, 2002.