APPROXIMATION WITH KRONECKER PRODUCTS

C.F. VAN LOAN and N. PITSIANIS
Department of Computer Science
Cornell University
Ithaca, New York 14853, U.S.A.
cv@cs.cornell.edu

ABSTRACT. Let A be an m-by-n matrix with $m = m_1 m_2$ and $n = n_1 n_2$. We consider the problem of finding $B \in \mathbb{R}^{m_1 \times n_1}$ and $C \in \mathbb{R}^{m_2 \times n_2}$ so that $\| A - B \otimes C \|_F$ is minimized. This problem can be solved by computing the largest singular value and associated singular vectors of a permuted version of A. If A is symmetric, definite, non-negative, or banded, then the minimizing B and C are similarly structured. The idea of using Kronecker product preconditioners is briefly discussed.

KEYWORDS. Kronecker product, preconditioners, block matrices.

1 Background

Suppose $A \in \mathbb{R}^{m \times n}$ with $m = m_1 m_2$ and $n = n_1 n_2$. This paper is about the minimization of

$$\phi_A(B, C) = \| A - B \otimes C \|_F^2$$

where $B \in \mathbb{R}^{m_1 \times n_1}$, $C \in \mathbb{R}^{m_2 \times n_2}$, and " $\otimes$ " denotes the Kronecker product.

Our interest in this problem stems from preliminary experience with Kronecker product preconditioners in the conjugate gradient setting. Suppose $A \in \mathbb{R}^{n \times n}$ with $n = n_1 n_2$ and that M is the preconditioner. For this solution process to be successful, the preconditioner should "capture" the essence of A as much as possible subject to the constraint that a linear system $Mz = r$ is "easy" to solve. In our context, we capture A through the minimization $\phi_A(B, C)$ with $B \in \mathbb{R}^{n_1 \times n_1}$ and $C \in \mathbb{R}^{n_2 \times n_2}$. Systems of the form $Mz \equiv (B \otimes C)z = r$ are easy to solve because only $O(n^{3/2})$ flops are required if $n_1 \approx n_2 \approx \sqrt{n}$. To appreciate this point, observe that $(B \otimes C)z = r$ is equivalent to

$$CZB^T = R \tag{1}$$

M. S. Moonen et al. (eds.), Linear Algebra for Large Scale and Real-Time Applications, 293–314.

where Z and R are n_2-by-n_1 matrices whose columns are segments of the vectors z and r respectively:

$$\begin{array}{l} Z(:,k) = z((k-1)n_2 + 1{:}kn_2) \\ \\ R(:,k) = r((k-1)n_2 + 1{:}kn_2) \end{array} \qquad k = 1{:}n_1 \,.$$

(At this point the reader may wish to review the algebra of Kronecker products. See [21] or [29].) If B and C are nonsingular and we apply Gaussian elimination with partial pivoting to produce the factorizations $P_1B = L_1U_1$ and $P_2C = L_2U_2$, then $2(n_1^3 + n_2^3)/3$ flops are required. The ensuing multiple triangular system solves involve an additional $2(n_1^2n_2+n_1n_2^2)$ flops. If $n = n_1^2 = n_2^2$, then a total of $16n^{3/2}/3$ flops are needed.

An instructive way to look at the above solution process is to recognize that

$$(P_1 \otimes P_2)(B \otimes C) \;=\; (L_1 \otimes L_2)(U_1 \otimes U_2)$$

is an LU (with partial pivoting) factorization of $B \otimes C$. This illustrates the adage that *a given factorization of $B \otimes C$ can usually be obtained by taking the Kronecker product of the corresponding B and C factorizations* :

$$\begin{array}{lll}
\text{Cholesky:} & \begin{array}{l} B = L_1L_1^T \\ C = L_2L_2^T \end{array} & \Rightarrow \;\; (B \otimes C) = (L_1 \otimes L_2)(L_1 \otimes L_2)^T \\[2ex]
\text{QR:} & \begin{array}{l} B = Q_1R_1 \\ C = Q_2R_2 \end{array} & \Rightarrow \;\; (B \otimes C) = (Q_1 \otimes Q_2)(R_1 \otimes R_2)^T \\[2ex]
\text{SVD:} & \begin{array}{l} B = U_1\Sigma_1V_1^T \\ C = U_2\Sigma_2V_2^T \end{array} & \Rightarrow \;\; (B \otimes C) = (U_1 \otimes U_2)(\Sigma_1 \otimes \Sigma_2)(V_1 \otimes V_2)^T \\[2ex]
\text{Schur:} & \begin{array}{l} B = U_1D_1U_1^H \\ C = U_2D_2U_2^H \end{array} & \Rightarrow \;\; (B \otimes C) = (U_1 \otimes U_2)(D_1 \otimes D_2)(U_1 \otimes U_2)^H
\end{array}$$

Here we are exploiting the fact that

$$\text{Kronecker products of} \left\{ \begin{array}{c} \text{orthogonal} \\ \text{triangular} \\ \text{diagonal} \end{array} \right\} \text{matrices are} \left\{ \begin{array}{c} \text{orthogonal} \\ \text{triangular} \\ \text{diagonal} \end{array} \right\}.$$

For a practical illustration of Kronecker product factorizations, see [10] where the idea is applied with QR in a photogrammetry application.

Some factorizations are not "preserved" when Kronecker products are taken:

- A real Schur decomposition of $B \otimes C$ is not obtained by taking the Kronecker product of the real Schur decompositions of B and C because the 2-by-2 bumps in the factors can create "block bumps" in the product. The computational ramifications of this fact are discussed in [3, 13].

- If QR with column pivoting is used to produce the factorizations $B\Pi_1 = Q_1R_1$ and $C\Pi_2 = Q_2R_2$, then $(B \otimes C)(\Pi_1 \otimes \Pi_2) = (Q_1 \otimes Q_2)(R_1 \otimes R_2)$ is *not* the factorization rendered by the same algorithm applied to $B \otimes C$.

Despite these anomalies, it is clear that the solution of Kronecker product systems is a nice problem with much structure to exploit. Not only are $O(n^{3/2})$ solution procedures available, but the form of (1) suggests opportunities for using the level-3 BLAS and parallel processing.

The act of finding good preconditioners through an appropriately constrained minimization of $\| A - M \|_F$ is not new. For example, [5] derives a useful class of preconditioners for the case when A is Toeplitz by solving

$$\min_{M \text{ circulant}} \quad \| A - M \|_F \quad .$$

Generalizations of this for matrices with Toeplitz blocks are discussed in [6].

Our presentation is organized as follows. First, we characterize the optimum Kronecker factors B and C in terms of the singular value decomposition of a permuted version of A. Algorithms for determining B and C are discussed is §3 and §4. The important cases when A is banded, non-negative, symmetric, and definite are handled in §5 along with some additional specially structured examples. In §6 we briefly examine the use of Kronecker product preconditioners.

We conclude this section with a few pointers to related work. The Kronecker product has a long history in mathematics and an excellent review is offered in [18]. Computational aspects of the operation are detailed in [25, 9].

Kronecker products arise in a number of applied areas. See [1, 28, 4, 17, 26] for Kronecker product discussions of generalized spectra, higher order statistics, systems theory, image processing, and photogrammetry.

In recent years there have been a number of developments that point to an increased role of the Kronecker product in the area of high performance matrix computations. In [22] is developed a parallel programming methodology that revolves around the Kronecker product. See also [23]. In [27, 29] is shown how the organization of fast transforms is clarified through the "language" of Kronecker products.

2 The rank-1 approximation

Consider the uniform blocking of an m_1m_2-by-n_1n_2 matrix A.

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1,n_1} \\ A_{21} & A_{22} & \cdots & A_{2,n_1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_1,1} & A_{m_1,2} & \cdots & A_{m_1,n_1} \end{bmatrix}, \qquad A_{ij} \in \mathbb{R}^{m_2 \times n_2} \,. \tag{2}$$

Using Matlab colon notation, the (i, j) block is given by

$$A_{ij} = A((i-1)m_2 + 1{:}im_2, (j-1)n_2 + 1{:}jn_2),$$

the submatrix defined by rows $(i-1)m_2 + 1$ to im_2 and columns $(j-1)n_2 + 1$ to jn_2. It is not hard to show using the definition of the Kronecker product that

$$\phi_A(B,C) = \sum_{i=1}^{m_1}\sum_{j=1}^{n_1} \| A_{ij} - b_{ij}C \|_F^2 . \tag{3}$$

By keeping the B matrix "intact," we also have

$$\phi_A(B,C) = \sum_{i=1}^{m_2}\sum_{j=1}^{n_2} \| \hat{A}_{ij} - c_{ij}B \|_F^2 , \tag{4}$$

where

$$\hat{A}_{ij} = A(i{:}m_2{:}m, j{:}n_2{:}n)$$

is the m_1-by-n_1 submatrix defined by rows i, $i + m_2$, $i + 2m_2$, ... , $i + (m_1 - 1)m_2$ and columns j, $j + n_2$, $j + 2n_2$, ... , $j + (n_1 - 1)n_2$. Thinking of matrices at the block level is the key to high performance matrix computations. See [15].

To proceed further with the analysis of $\phi_A(B, C)$, we require the *vec* operation, which is a way of turning matrices into vectors by "stacking" the columns:

$$X \in \mathbb{R}^{p\times q} \Rightarrow vec(X) = \begin{bmatrix} X(1{:}p,1) \\ X(1{:}p,2) \\ \vdots \\ X(1{:}p,q) \end{bmatrix} \in \mathbb{R}^{pq}.$$

It turns out that the *vec* operator can be used to express the minimization of $\| A - B \otimes C \|_F^2$ as a rank-1 approximation problem. The idea is to rearrange A into another matrix $\tilde{A}$ so that the sum of squares that arise in $\| A - B \otimes C \|_F^2$ is exactly the same as the sum of squares that arise in $\| \tilde{A} - vec(B)vec(C)^T \|_F^2$. For example, in a 4-by-4 problem with 2-by-2 blocks,

$$\| A - B \otimes C \|_F = \left\| \left[\begin{array}{cc|cc} a_{11} & a_{21} & a_{12} & a_{22} \\ a_{31} & a_{41} & a_{32} & a_{42} \\ \hline a_{13} & a_{23} & a_{14} & a_{24} \\ a_{33} & a_{43} & a_{34} & a_{44} \end{array}\right] - \begin{bmatrix} b_{11} \\ b_{21} \\ b_{12} \\ b_{22} \end{bmatrix} [c_{11} \;\; c_{21} \;\; c_{12} \;\; c_{22}] \right\|_F .$$

Refer to the above permuted version of A as $\tilde{A}$. Note that $\tilde{A}$ is *not* of the form PAQ where P and Q are permutation matrices. Indeed, in our example

- the four rows of $\tilde{A}$ are *vec*'s of the 2-by-2 blocks of A:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad \Rightarrow \quad \tilde{A} = \begin{bmatrix} vec(A_{11})^T \\ vec(A_{21})^T \\ vec(A_{12})^T \\ vec(A_{22})^T \end{bmatrix}.$$

- the *vec*'s of the 2-by-2 blocks of $\tilde{A}^T$ are columns of A:

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \quad \Rightarrow \quad A = \left[vec(\tilde{A}_{11}^T) \mid vec(\tilde{A}_{12}^T) \mid vec(\tilde{A}_{21}^T) \mid vec(\tilde{A}_{22}^T) \right].$$

In general, if $m = m_1 m_2$, $n = n_1 n_2$, $A \in \mathbb{R}^{m \times n}$, and we have the blocking (2), then we define the *rearrangement* of A (relative to the blocking parameters m_1, m_2, n_1, and n_2) by

$$\mathcal{R}(A) = \begin{bmatrix} A_1 \\ \vdots \\ A_{n_1} \end{bmatrix}, \qquad A_j = \begin{bmatrix} vec(A_{1,j})^T \\ \vdots \\ vec(A_{m_1,j})^T \end{bmatrix}, \qquad j = 1{:}n_1. \tag{5}$$

Note that $\mathcal{R}(A)$ has $m_1 n_1$ rows and $m_2 n_2$ columns. Thus, $\mathcal{R}(A)$ need not be the same size as A. For example, if $m = m_1 m_2 = 2{\cdot}2$ and $n = n_1 n_2 = 3{\cdot}2$, then A is 4-by-6 but

$$\mathcal{R}(A) = \left[\begin{array}{cc|cc} a_{11} & a_{21} & a_{12} & a_{22} \\ a_{31} & a_{41} & a_{32} & a_{42} \\ \hline a_{13} & a_{23} & a_{14} & a_{24} \\ a_{33} & a_{43} & a_{34} & a_{44} \\ \hline a_{15} & a_{25} & a_{16} & a_{26} \\ a_{35} & a_{45} & a_{36} & a_{46} \end{array}\right].$$

We are now set to establish a key result that connects the problem of minimizing $\phi_A(B, C)$ with the problem of approximating $\tilde{A}$ with a rank-1 matrix.

Theorem 1 *Assume that $A \in \mathbb{R}^{m \times n}$ with $m = m_1 m_2$ and $n = n_1 n_2$. If $B \in \mathbb{R}^{m_1 \times n_1}$ and $C \in \mathbb{R}^{m_2 \times n_2}$, then*

$$\| A - B \otimes C \|_F = \| \mathcal{R}(A) - vec(B) vec(C)^T \|_F.$$

Proof. By applying the *vec* operator in (3) we get:

$$\begin{aligned} \| A - B \otimes C \|_F^2 &= \sum_{j=1}^{n_1} \sum_{i=1}^{m_1} \| vec(A_{ij}) - b_{ij} vec(C) \|_2^2 \\ &= \sum_{j=1}^{n_1} \sum_{i=1}^{m_1} \| vec(A_{ij})^T - b_{ij} vec(C)^T \|_2^2 \\ &= \sum_{j=1}^{n_1} \| A_j - B(:, j) vec(C)^T \|_F^2 \\ &= \| \mathcal{R}(A) - vec(B) vec(C)^T \|_F^2. \quad \Box \end{aligned}$$

The approximation of a given matrix by a rank-1 matrix has a well-known solution in terms of the singular value decomposition.

Corollary 2 *Assume that $A \in \mathbb{R}^{m \times n}$ with $m = m_1 m_2$ and $n = n_1 n_2$. If $\tilde{A} = \mathcal{R}(A)$ has singular value decomposition*

$$U^T \tilde{A} V = \Sigma = \mathrm{diag}(\sigma_i)$$

where σ_1 is the largest singular value, and $U(:,1)$ and $V(:,1)$ are the corresponding singular vectors, then the matrices $B \in \mathbb{R}^{m_1 \times n_1}$ and $C \in \mathbb{R}^{m_2 \times n_2}$ defined by $vec(B) = \sigma_1 U(:,1)$ and $vec(C) = V(:,1)$ minimize $\| A - B \otimes C \|_F$.

Proof. See [15, p. 73]. □

The definition (5) of $\mathcal{R}(A)$ is in terms of the blocks A_{ij} in (2). An alternative characterization can be obtained in terms of the columns of A. In particular, we show that

$$\mathcal{R}(A) = \begin{bmatrix} \tilde{A}_{11} & \cdots & \tilde{A}_{1,n_2} \\ \vdots & \ddots & \vdots \\ \tilde{A}_{n_1,1} & \cdots & \tilde{A}_{n_1,n_2} \end{bmatrix}. \tag{6}$$

where $\tilde{A}_{ij} \in \mathbb{R}^{m_1 \times m_2}$ is defined by

$$vec(\tilde{A}_{ij}^T) = A(:,(i-1)n_2 + j) \qquad 1 \le i \le n_1,\ 1 \le j \le n_2\,.$$

In view of (5) we need only confirm that

$$A_i = \begin{bmatrix} vec(A_{1,i})^T \\ \vdots \\ vec(A_{m_1,i})^T \end{bmatrix} = \left[\ \tilde{A}_{i,1} \mid \tilde{A}_{i,2} \mid \cdots \mid \tilde{A}_{i,n_2}\ \right]. \tag{7}$$

For $s = 1{:}m_2$, $p = 1{:}n_2$, and $q = 1{:}m_2$ we have

$$[A_i]_{s,(p-1)m_2+q} = \left[vec(A_{s,i})^T\right]_{(p-1)m_2+q} = A((s-1)m_2 + q, (i-1)n_2 + p)\,.$$

But (7) immediately follows because we also have

$$\left[\ \tilde{A}_{i,1} \mid \tilde{A}_{i,2} \mid \cdots \mid \tilde{A}_{i,n_2}\ \right]_{s,(p-1)m_2+q} = \left[\tilde{A}_{i,p}\right]_{sq} = A((s-1)m_2 + q, (i-1)n_2 + p)\,.$$

3 SVD framework

The Golub-Reinsch SVD algorithm can be used for computing the largest singular value and corresponding singular vectors of $\mathcal{R}(A)$. However, in view of the potentially large dimension of $\tilde{A} = \mathcal{R}(A)$ in some applications, it may be more appropriate to use the SVD Lanczos process described in [12]. Here is how to proceed with the computation of $B \in \mathbb{R}^{m_1 \times n_1}$ and $C \in \mathbb{R}^{m_2 \times n_2}$:

Framework 1.

C = initial guess.
$v_1 \leftarrow vec(C)/\| C \|_F$
$p_0 \leftarrow v_1;\ \beta_0 \leftarrow 1;\ j \leftarrow 0;\ u_0 \leftarrow 0$
while $\beta_j \neq 0$ (or some other less stringent criteria.)
    $v_{j+1} \leftarrow p_j/\beta_j$
    $j \leftarrow j+1$
    $r_j \leftarrow \tilde{A}v_j - \beta_{j-1}u_{j-1}$
    $\alpha_j \leftarrow \| r_j \|_2$
    $u_j \leftarrow r_j/\alpha_j$
    $p_j \leftarrow \tilde{A}^T u_j - \alpha_j v_j;$
    $\beta_j \leftarrow \| p_j \|_2$
    end {while}
Compute the largest singular value σ_1 and associated left and right
    singular vectors u_B and v_B of the bidiagonal matrix with diagonal
    $\alpha_1, \ldots, \alpha_j$ and upper diagonal $\beta_1, \ldots, \beta_{j-1}$.
Define B by $vec(B) = \sigma_1[u_1, \ldots, u_j]u_B$ and C by $vec(C) = [v_1, \ldots, v_j]v_B$

There are many subtleties associated with the Lanzcos process and we refer the reader to [8] or [15, p. 98ff] for details.

Our only implementation discussion concerns the matrix-vector products $\tilde{A}x$ and $\tilde{A}^T x$ that are required by the iteration. The explicit formation of $\mathcal{R}(A) = \tilde{A}$ is *not* necessary. For example, working with the characterization (5), here is a dot product formulation for $y \leftarrow \tilde{A}x$:

for $j = 1{:}n_1$
    for $i = 1{:}m_1$
        $y((j-1)m_1 + i) \leftarrow vec(A_{ij})^T x$
    end
end

A saxpy-based procedure for $y \leftarrow \tilde{A}^T x$ proceeds as follows:

$y(1{:}m_2n_2) \leftarrow 0$
for $j = 1{:}n_1$
    for $i = 1{:}m_1$
        $y \leftarrow y + x((j-1)m_1 + i)vec(A_{ij})$
    end
end

By working with (6) we have the following alternative block formulation for $y \leftarrow \tilde{A}x$:

$y(1{:}m_1n_1) \leftarrow 0$
for $i = 1{:}n_1$
    $rows = (i-1)m_1 + 1{:}im_1$
    for $j = 1{:}n_2$
        Define $Z \in \mathbb{R}^{m_1 \times m_2}$ by $vec(Z^T) = A(:, (i-1)n_2 + j)$
        $y(rows) \leftarrow y(rows) + Zx((j-1)m_2 + 1{:}jm_2)$

end
end

Likewise, we can formulate a procedure for $y \leftarrow \tilde{A}^T x$ that is based upon (6):

$y(1{:}m_2n_2) \leftarrow 0$
for $i = 1{:}n_2$
  $rows = (i-1)m_2 + 1{:}im_2$
  for $j = 1{:}n_1$
    Define $Z \in \mathbb{R}^{m_2 \times m_1}$ by $vec(Z^T) = A(:, (j-1)n_2 + i)$
    $y(rows) \leftarrow y(rows) + Z^T x((j-1)m_1 + 1{:}jm_1)$
  end
end

Each of these products requires $2m_1n_1m_2n_2 = 2mn$ flops assuming that $\tilde{A}$ is treated as a dense matrix.

4 The separable least squares framework

Note that if we fix C, then the problem of minimizing $\phi_A(B, C) = \| A - B \otimes C \|_F$ is a linear least squares problem with unknowns b_{ij}. Likewise, if B is fixed, then the minimization of ϕ_A is a linear least squares problem in the c_{ij}. The following theorem specifies the solution to these linear least squares problems and requires the concept of matrix trace:

$$X \in \mathbb{R}^{q \times q} \;\Rightarrow\; tr(X) = \sum_{i=1}^{q} x_{ii} \,.$$

Theorem 3 *Suppose $m = m_1m_2$, $n = n_1n_2$, and $A \in \mathbb{R}^{m \times n}$. If $C \in \mathbb{R}^{m_2 \times n_2}$ is fixed, then the matrix $B \in \mathbb{R}^{m_1 \times n_1}$ defined by*

$$b_{ij} \;=\; \frac{tr(A_{ij}^T C)}{tr(C^T C)} \qquad 1 \le i \le m_1,\ 1 \le j \le n_1 \tag{8}$$

minimizes $\| A - B \otimes C \|_F$ where $A_{ij} = A((i-1)m_2 + 1{:}im_2, (j-1)n_2 + 1{:}jn_2)$. Likewise, if $B \in \mathbb{R}^{m_1 \times n_1}$ is fixed, then the matrix $C \in \mathbb{R}^{m_2 \times n_2}$ defined by

$$c_{ij} \;=\; \frac{tr(\hat{A}_{ij}^T B)}{tr(B^T B)} \qquad 1 \le i \le m_2,\ 1 \le j \le n_2 \tag{9}$$

minimizes $\| A - B \otimes C \|_F$ where $\hat{A}_{ij} = A(i{:}m_2{:}m, j{:}n_2{:}n)$.

Proof. Since

$$\begin{aligned} \| A_{ij} - b_{ij}C \|_F^2 &= tr((A_{ij} - b_{ij}C)^T (A_{ij} - b_{ij}C)) \\ &= \| A_{ij} \|_F^2 - 2b_{ij} tr(C^T A_{ij}) + b_{ij}^2 \| C \|_F^2 \end{aligned}$$

it follows from (2.2) that

$$\frac{\partial \phi_A(B, C)}{\partial b_{ij}} \;=\; -2\, tr(C^T A_{ij}) + 2b_{ij} \| C \|_F^2 \,.$$

Setting these partials to zero defines the required matrix B. The proof of (9) is similar. □

The above result suggests that we can compute B and C by taking the *separable least squares* described in [2]. The idea is to minimize $\phi_{\mathcal{A}}(B,C)$ by alternately improving the B and C matrices through a sequence of linear least squares optimizations:

Framework 2.

$C = C_0$ (given starting matrix)
Repeat:
$\quad\gamma \leftarrow tr(C^TC)$
$\quad$**for** $i = 1{:}m_1$
$\quad\quad$**for** $j = 1{:}n_1$
$\quad\quad\quad b_{ij} \leftarrow tr(C^TA_{ij})/\gamma$
$\quad\beta \leftarrow tr(B^TB)$
$\quad$**for** $i = 1{:}m_2$
$\quad\quad$**for** $j = 1{:}n_2$
$\quad\quad\quad c_{ij} \leftarrow tr(B^T\hat{A}_{ij})/\beta$

This process requires $4m_1n_1m_2n_2 = 4mn$ flops per iteration, the same as Framework 1. Other methods for nonlinear least squares problems with variables that separate are discussed in [14, 24].

Framework 2 amounts to a power method for the largest singular value of $\tilde{A} = \mathcal{R}(\mathcal{A})$. To see this we switch to "tilde-space" and observe that if

$$\phi(b,c) \;=\; \| \tilde{A} - bc^T \|_F^2 \qquad b \in \mathbb{R}^{m_1n_1},\; c \in \mathbb{R}^{m_2n_2},$$

then the gradient is given by

$$\nabla\phi(b,c) \;=\; -2\begin{bmatrix} \tilde{A}c - (c^Tc)b \\ \tilde{A}^Tb - (b^Tb)c \end{bmatrix}.$$

If b is fixed, then the minimizing c is obtained by setting $c = \tilde{A}^Tb/b^Tb$ for then the c-partials are all zero. Likewise, if c is fixed, then the minimizing b is given by $b = \tilde{A}c/c^Tc$. After k passes through the iteration

$c = c_0$ (given starting vector)
Repeat:
$\quad b \leftarrow \tilde{A}c/c^Tc$
$\quad c \leftarrow \tilde{A}^Tb/b^Tb$

the vector c is in the direction of $(\tilde{A}^T\tilde{A})^kc_0$ and the vector b is in the direction of $(\tilde{A}\tilde{A}^T)^{k-1}\tilde{A}c_0$.

The practical implementation of this framework involves all the subtleties that are associated with the power method. See [30] for a discussion.

5 Structured problems

As we alluded to in §1, the Kronecker product of two structured matrices is usually structured in the same way:

$$\text{If } B \text{ and } C \text{ are } \left\{ \begin{array}{l} \text{banded} \\ \text{non-negative} \\ \text{symmetric} \\ \text{positive definite} \\ \text{stochastic} \\ \text{orthogonal} \end{array} \right\}, \text{ then } B \otimes C \text{ is } \left\{ \begin{array}{l} \text{banded} \\ \text{non-negative} \\ \text{symmetric} \\ \text{positive definite} \\ \text{stochastic} \\ \text{orthogonal} \end{array} \right\}.$$

We are interested in the structure of the solution to the Kronecker approximation problem given that A is structured. In the following subsections we use Corollary 2 and Theorem 3 to establish a number results about structured problems.

5.1 BANDEDNESS

We first show how bandedness in A "shows up" in B and C.

Theorem 4 *Suppose $n = n_1 n_2$, $A \in \mathbb{R}^{n \times n}$ has bandwidth pn_2, and that each block in*

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1,n_1} \\ \vdots & \ddots & \vdots \\ A_{n_1,1} & \cdots & A_{n_1 n_1} \end{bmatrix} \qquad A_{ij} \in \mathbb{R}^{n_2 \times n_2}$$

has bandwidth q or less. If $B \in \mathbb{R}^{n_1 \times n_1}$ and $C \in \mathbb{R}^{n_2 \times n_2}$ minimize $\| A - B \otimes C \|_F$, then B has bandwidth p and C has bandwidth q.

Proof. Since A has bandwidth pn_2, it follows that $A_{ij} = 0$ if $|i - j| > p$. From (3) we have $b_{ij} = 0$ whenever $|i - j| > p$. Since each A_{ij} has bandwidth q, it follows that the minimization of $\| A_{ij} - b_{ij} C \|_F$ requires setting c_{rs} to zero whenever $|r - s| > q$. Thus, a minimizing C must have bandwidth q. □

5.2 NON-NEGATIVITY

We first show that if A and C are non-negative, then the B that minimizes $\phi_A(B, C)$ is also non-negative.

Theorem 5 *If $m = m_1 m_2$, $n = n_1 n_2$, $A \in \mathbb{R}^{m \times n}$, and $C \in \mathbb{R}^{m_2 \times n_2}$ are non-negative, then there exists a non-negative $B \in \mathbb{R}^{m_1 \times n_1}$ that minimizes $\| A - B \otimes C \|_F$.*

Proof. Using the non-negativity of C and Theorem 3,

$$b_{ij} = \frac{tr(A_{ij}^T C)}{tr(C^T C)} \geq 0$$

for $i = 1{:}m_1$ and $j = 1{:}n_1$. □

In the same way, we can show that if A and B are non-negative, then the C that minimizes $\| A - B \otimes C \|$ is also non-negative. Thus, if we start with a non-negative C in Framework 2, then all subsequent B and C matrices are non-negative. The following theorem shows that this restriction poses no difficultly because the optimum B and C are also non-negative.

Theorem 6 *If $m = m_1m_2$, $n = n_1n_2$, and $A \in \mathbb{R}^{m\times n}$ is non-negative, then there exist non-negative matrices $B \in \mathbb{R}^{m_1\times n_1}$ and $C \in \mathbb{R}^{m_2\times m_2}$ such that $\| A - B \otimes C \|_F$ is minimized.*

Proof. Note that $\tilde{A} = \mathcal{R}(A)$ has non-negative entries and let σ_1 be its largest singular value. Peron-Frobenius theory tells us that there exist non-negative $u \in \mathbb{R}^{m_1n_1}$ and $v \in \mathbb{R}^{m_2n_2}$ so that $\tilde{A}^T\tilde{A}v = \sigma_1^2 v$ and $\tilde{A}\tilde{A}^T u = \sigma_1^2 u$. (See [20, p.503].) But u and v are the right and left singular vectors of $\tilde{A}$ and so the matrices B and C as specified in Corollary 2 are non-negative. □

5.3 SYMMETRY

Turning next to the issue of symmetry, we show that if A and C are symmetric, then a symmetric B can be found to minimize $\phi_A(B, C)$.

Theorem 7 *If $n = n_1n_2$, $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n_2\times n_2}$ are symmetric, then there exists a symmetric $B \in \mathbb{R}^{n_1\times n_1}$ that minimizes $\| A - B \otimes C \|_F$.*

Proof. Since A is symmetric, $A_{ji} = A_{ij}^T$. Using elementary properties of the trace we have

$$b_{ij} = \frac{tr(A_{ij}^T C)}{tr(C^T C)} = \frac{tr(A_{ji} C)}{tr(C^T C)} = \frac{tr(C A_{ji}}{tr(C^T C)} = \frac{tr(A_{ji}^T C)}{tr(C^T C)} = b_{ji}$$

for all $1 \le i, j \le n_1$. It follows that B is symmetric. □

It is equally straightforward to establish that a symmetric C can be found to minimize $\| A - B \otimes C \|_F$ is A and B are symmetric.

Analogous results are applicable if the "frozen factor" is skew-symmetric:

Theorem 8 *If $n = n_1n_2$, $A \in \mathbb{R}^{n\times n}$ is symmetric and $C \in \mathbb{R}^{n_2\times n_2}$ is skew-symmetric, then there exists a skew-symmetric $B \in \mathbb{R}^{n_1\times n_1}$ that minimizes $\| A - B \otimes C \|_F$.*

Proof.

$$b_{ij} = \frac{tr(A_{ij}^T C)}{tr(C^T C)} = -\frac{tr(A_{ji}^T C)}{tr(C^T C)} = -b_{ji}\,. \qquad \square$$

The optimum Kronecker approximation of a symmetric matrix may have skew-symmetric factors as consideration of the following example shows:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

For this particular A, it is not possible to find symmetric B and C for which we have $A = B \otimes C$. The following theorem summarizes the situation.

Theorem 9 *Suppose $n = n_1n_2$ and $A \in \mathbb{R}^{n\times n}$ is symmetric. If $\| A - B \otimes C \|_F$ cannot minimized by symmetric matrices $B \in \mathbb{R}^{n_1\times n_1}$ and $C \in \mathbb{R}^{n_2\times n_2}$, then it can be minimized by skew-symmetric matrices $B \in \mathbb{R}^{n_1\times n_1}$ and $C \in \mathbb{R}^{n_2\times n_2}$.*

Proof. For any positive integer q, define the following orthogonal subpaces of $\mathbb{R}^{q^2}$:

$$S_+^{(q)} = \{x \in \mathbb{R}^{q^2} : x = vec(X) \text{ for some symmetric } X \in \mathbb{R}^{q\times q}\}$$

$$S_-^{(q)} = \{x \in \mathbb{R}^{q^2} : x = vec(X) \text{ for some skew-symmetric } X \in \mathbb{R}^{q\times q}\}$$

Note that $\mathbb{R}^{q^2} = S_+^{(q)} \oplus S_1^{(q)}$.

Now suppose that $y = \mathcal{R}(A)x$ and that $X \in \mathbb{R}^{n_2\times n_2}$ and $Y \in \mathbb{R}^{n_1\times n_1}$ are defined by $x = vec(X)$ and $y = vec(Y)$, respectively. From (2)) we know that

$$[Y]_{ij} = vec(A_{ij})^T x = tr(A_{ij}^T X) \qquad 1 \le i,j \le n_1 .$$

If $x \in S_+^{(n_2)}$, then since A is symmetric we have

$$[Y]_{ij} - [Y]_{ji} = tr((A_{ij}^T - A_{ji}^T)X) = tr((A_{ij}^T - A_{ij})X) = vec(A_{ij}^T - A_{ij})^T x = 0$$

since $vec(A_{ij}^T - A_{ij}) \in S_-^{(n_2)}$. Thus,

$$x \in S_+^{(n_2)} \Rightarrow \mathcal{R}(A)x \in S_+^{(n_1)}$$

Likewise,

$$x \in S_1^{(n_2)} \Rightarrow \mathcal{R}(A)x \in S_1^{(n_1)}$$

and so $(S_+^{(n_2)}, S_+^{(n_1)})$ and $(S_-^{(n_2)}, S_-^{(n_1)})$ are singular subspace pairs for $\mathcal{R}(A)$. It follows that the largest singular value and corresponding singular vectors must be associated with one of these pairs. □

Theorem 9 can also be established by observing that if A is symmetric, then

$$P_{n_1}\mathcal{R}(A)P_{n_2}^T = \mathcal{R}(A)$$

where P_q designates the *vec permutation matrix* on $\mathbb{R}^{q^2}$:

$$P_q vec(X) = vec(X^T) \qquad X \in \mathbb{R}^{q\times q}.$$

This permutation connects the *vec* of a matrix and the *vec* of its transpose. See [19] for further details.

5.4 POSITIVE DEFINITENESS

We first show that if the initial guess matrix in Framework 2 is positive definite, then all subsequent B and C iterates are positive definite.

Theorem 10 *If $n = n_1^2$, $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n_2\times n_2}$ are symmetric positive definite, then there exists a symmetric positive definite $B \in \mathbb{R}^{n_1\times n_1}$ that minimizes $\phi_A(B,C)$. Likewise, if $B \in \mathbb{R}^{n_1\times n_1}$ is symmetric positive definite, then there exists a symmetric positive definite $C \in \mathbb{R}^{n_2\times n_2}$ that minimizes $\phi_A(B,C)$.*

Proof. If each entry b_{ij} in $B \in \mathbb{R}^{n_1 \times n_1}$ satisfies $b_{ij} = tr(C^T A_{ij})/tr(C^T C)$, and if $y \in \mathbb{R}^{n_1}$, then using the linearity of the trace we have

$$y^T B y \;=\; \sum_{i=1}^{n_1}\sum_{j=1}^{n_1} b_{ij} y_i y_j \;=\; \sum_{i=1}^{n_1}\sum_{j=1}^{n_1} y_i y_j tr(C^T A_{ij})/tr(C^T C) \;=\; tr(C^T \hat{A}) \tag{10}$$

where

$$\hat{A} \;=\; \sum_{i=1}^{n_1}\sum_{j=1}^{n_1} y_i y_j A_{ij}\,.$$

The matrix $\hat{A}$ is positive definite because for any $z \in \mathbb{R}^{n_1}$ we have

$$0 < (z \otimes y)^T A (z \otimes y) \;=\; \left[\, z_1 y^T \mid \cdots \mid z_{n_1} y^T \,\right] [A_{ij}] \begin{bmatrix} z_1 y \\ \vdots \\ z_{n_1} y \end{bmatrix} \;=\; z^T \hat{A} z\,.$$

Since C is positive definite, it has a Cholesky factorization $C = LL^T$. From (10) and the fact that the trace is invariant under similarity transformations, gives

$$y^T B y \;=\; tr(C^T \hat{A}) \;=\; tr(LL^T \hat{A}) \;=\; tr(L^{-1}(LL^T \hat{A})L) \;=\; tr(L^T \hat{A} L) \;>\; 0.$$

The proof that C is positive definite when B is given is similar. □

The next result shows that if A is symmetric and positive definite, then the same can be said about the optimum B and C.

Theorem 11 *If $n = n_1 n_2$ and $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, then there exists symmetric positive definite $B \in \mathbb{R}^{n_1 \times n_1}$ and $C \in \mathbb{R}^{n_2 \times n_2}$ that minimize $\phi_A(B, C)$.*

Proof. From Theorem 9 we may select the optimum B and C to be either both skew-symmetric or both symmetric. We first show that the latter must be the case.

If B is skew-symmetric, then there exists a real orthogonal U_B such that

$$U_B^T B U_B \;=\; B_1 \tag{11}$$

where B_1 is a direct sum of 1-by-1 and 2-by-2 skew-symmetric blocks. The 1-by-1's are (of course) zero and the 2-by-2's have the form

$$M \;=\; \begin{bmatrix} 0 & m \\ -m & 0 \end{bmatrix}$$

and correspond to the complex conjugate eigenpairs of B. The decomposition (11) is just the real Schur decomposition. Note that the unitary matrix

$$Z \;=\; \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$$

diagonalizes M:

$$Z^H M Z = \begin{bmatrix} im & 0 \\ 0 & -im \end{bmatrix}.$$

Let V_B be the unitary matrix that has copies of Z on the diagonal which correspond to the 2-by-2 blocks in B_1, and which is the identity elsewhere. It follows that

$$V_B^H U_B^T B U_B V_B = D_B$$

is diagonal. Let us refer to this decomposion as the ***structured Schur decomposition*** of B. Assume that C is also skew-symmetric and let

$$V_C^H U_C^T C U_C V_C = D_C$$

be its structured Schur decomposition. For a matrix H, let $|H|$ be the matrix obtained by taking the absolute values of each entry. Since

$$Z \left| \begin{bmatrix} im & 0 \\ 0 & -im \end{bmatrix} \right| Z^H = |m| I_2$$

it is easy to check that the matrices

$$\begin{aligned} B_+ &= U_B V_B \, |D_B| \, V_B^H U_B^T \\ C_+ &= U_C V_C \, |D_C| \, V_C^H U_C^T \end{aligned}$$

are real and symmetric.

Let $Q = Q_B \otimes Q_C$ where $Q_B = U_B V_B$ and $Q_C = U_C V_C$. Define the *off* operation on matrices as follows:

$$\mathit{off}(M) = \sum_{i \neq j} m_{ij}^2 .$$

Setting D_A to be the diagonal part of $Q^H A Q$, we see that

$$\begin{aligned} \| A - B_+ \otimes C_+ \|_F^2 &= \| Q^H A Q - |D_B| \otimes |D_C| \, \|_F^2 \\ &= \mathit{off}(Q^H A Q) + \| D_A - |D_B| \otimes |D_C| \, \|_F^2 \end{aligned}$$

while

$$\begin{aligned} \| A - B \otimes C \|_F^2 &= \| Q^H A Q - D_B \otimes D_C \|_F^2 \\ &= \mathit{off}(Q^H A Q) + \| D_A - D_B \otimes D_C \|_F^2 . \end{aligned}$$

Since $Q^H A Q$ is positive definite, D_A has positive diagonal entries. Moreover, $D_B \otimes D_C$ is a real diagonal matrix with some negative diagonal entries. It follows that

$$\| D_A - |D_B| \otimes |D_C| \, \|_F^2 < \| D_A - D_B \otimes D_C \|_F^2 .$$

and so

$$\| A - B_+ \otimes C_+ \|_F < \| A - B \otimes C \|_F .$$

This shows that a skew-symmetric pair cannot minimize $\phi_A(B, C)$.

Knowing now that the optimizing B and C are symmetric, it remains for us to show that they are both positive definite. Suppose

$$\begin{aligned} Q_1^T B Q_1 &= D_1 = \text{diag}(\lambda_1, \ldots, \lambda_{n_1}) \\ Q_2^T C Q_2 &= D_2 = \text{diag}(\mu_1, \ldots, \mu_{n_2}) \end{aligned}$$

are Schur decompositions. Set $Q = Q_1 \otimes Q_2$ and let $D=$ diag$(d_1, \ldots, d_n)$ be the diagonal part of $F = Q^T A Q$. Thus,

$$\begin{aligned} \| A - B \otimes C \|_F^2 &= \| Q^T(A - B \otimes C)Q \|_F^2 \\ &= \| F - D_1 \otimes D_2 \|_F^2 = \| D - D_1 \otimes D_2 \|_F^2 + \mathit{off}(F). \end{aligned}$$

Note that

$$\| D - D_1 \otimes D_2 \|_F^2 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} = (d_{(i-1)n_2+j} - \lambda_i \mu_j)^2 .$$

Since D has positive diagonal entries and

$$(d_{(i-1)n_2+j} - \lambda_i \mu_j)^2 - (d_{(i-1)n_2+j} - |\lambda_i \mu_j|)^2 = |\lambda_i|^2 |\mu_j|^2 - \lambda_i^2 \mu_j^2, > 0,$$

it follows that the λ_i and μ_j should all have the same sign. Otherwise, B and C will not render the minimum sum of squares. Since $\phi_A(-B, -C) = \phi_A(B, C)$, we may assume without loss of generality that this sign is positive. This implies that symmetric positive definite B and C may be chosen to be minimize $\phi_A(B, C)$. □

5.5 SUMS OF KRONECKER PRODUCTS

Next, we consider the situation when the matrix A to be approximated is a sum of Kronecker products:

$$A = \sum_{i=1}^{p} (G_i \otimes F_i) .$$

Assume that each G_i is m_1-by-n_1 and each F_i is m_2-by-n_2. It follows that if $f_i = vec(F_i)$ and $g_i = vec(G_i)$, then

$$\tilde{A} = \mathcal{R}(A) = \sum_{i=1}^{p} \mathcal{R}(G_i \otimes F_i) = \sum_{i=1}^{p} g_i f_i^T$$

is a rank-p matrix. This has two important ramifications. First, it means that matrix-vector products of the form $\tilde{A}x$ and $\tilde{A}^T x$ cost $O((m+n)p)$ flops where $m = m_1 m_2$ and $n = n_1 n_2$. Second, it means that the optimum B and C are linear combinations of the G_i and F_i:

$$\begin{aligned} B &= \alpha_1 G_1 + \cdots + \alpha_p G_p \\ C &= \beta_1 F_1 + \cdots + \beta_p F_p \end{aligned}$$

The problem of approximating matrices of the form $(I \otimes F) + (G \otimes I)$ is discussed further in §6.

5.6 APPROXIMATION WITH LINEAR HOMOGENEOUS CONSTRAINTS

Consider the problem of approximating A with a Kronecker product $B \otimes C$ that has a prescribed structure. If the constraints on B and C are linear and homogeneous, then we are looking at a problem with the following form:

$$\begin{array}{ll} \min & \| A - B \otimes C \|_F \, . \\ S_1^T vec(B) = 0 & \\ S_2^T vec(C) = 0 & \end{array} \tag{12}$$

Here, $A \in \mathbb{R}^{m \times n}$, $m = m_1 m_2$, $n = n_1 n_2$, $B \in \mathbb{R}^{m_1 \times n_1}$, $C \in \mathbb{R}^{m_2 \times n_2}$, $S_1 \in \mathbb{R}^{m_1 n_1 \times p_1}$, $S_2 \in \mathbb{R}^{m_2 n_2 \times p_2}$, and we assume that S_1 and S_2 have full column rank. By choosing these constraint matrices properly, we can force B and C to take on any prescribed sparsity pattern. Circulant, Toeplitz, Hankel, and Hamiltonian structures can also be imposed.

To solve the constrained problem we follow the techniques espoused in [11] where various modified eigenvalue problems are discussed. Let $b = vec(B)$, $c = vec(C)$, and assume that we have the QR factorizations

$$S_1 = Q_1 \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \qquad S_2 = Q_2 \begin{bmatrix} R_2 \\ 0 \end{bmatrix} \tag{13}$$

where R_1 and R_2 are square. If

$$Q_1^T \mathcal{R}(A) Q_2 = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}, \qquad Q_1^T b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \qquad Q_2^T c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

are partitioned conformably with (13), then (12) transforms to the problem of minimizing

$$\left\| \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}^T \right\|_F$$

subject to the constraints

$$\begin{bmatrix} R_1^T \,|\, 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = 0, \qquad \begin{bmatrix} R_2^T \,|\, 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0.$$

It follows that b_1 and c_1 are both zero and that the optimum b_2 and c_2 can be obtained by solving the unconstrained problem

$$\min \| \tilde{A}_{22} - b_2 c_2^T \|_F \, .$$

Collecting results, we see that B and C are prescribed by

$$vec(B) = Q_1 \begin{bmatrix} 0 \\ b_2 \end{bmatrix}, \qquad vec(C) = Q_2 \begin{bmatrix} 0 \\ c_2 \end{bmatrix}.$$

5.7 STOCHASTIC AND ORTHOGONAL PROBLEMS

The non-negative matrix $A \in \mathbb{R}^{n \times n}$ is *stochastic* if $e_n^T A = e_n^T$ where e_n is the n-vector of ones. If $n = n_1 n_2$ and $B \in \mathbb{R}^{n_1 \times n_1}$ and $C \in \mathbb{R}^{n_2 \times n_2}$ minimize $\phi_A(B, C)$, then it does *not* follow that B and C are stochastic. For example, if

$$A = \begin{bmatrix} .1 & .5 & .2 & .6 \\ .4 & .1 & .1 & .2 \\ .2 & .0 & .3 & .1 \\ .3 & .4 & .4 & .1 \end{bmatrix},$$

then, after normalizing B and C so that $b_{11} + b_{21} = 1$ we have

$$B = \begin{bmatrix} .6228 & .5939 \\ .3772 & .4298 \end{bmatrix} \qquad C = \begin{bmatrix} .3610 & .6657 \\ .5560 & .3512 \end{bmatrix}.$$

Note that B and C are not quite stochastic. Thus, to get the best stochastic Kronecker product approximation we must apply a constrained nonlinear least squares solver to the problem

$$\min_{\substack{e_{n_1}^T B = e_{n_1}^T,\ B \geq 0 \\ e_{n_2}^T C = e_{n_2}^T,\ C \geq 0}} \| A - B \otimes C \|_F$$

Another structured problem that is not solvable by our SVD framework is the case when A is orthogonal and we insist that the optimizing B and C be orthogonal. It does *not* follow that orthogonal B and C minimize $\phi_A(B,C)$. Thus, we are led to another constrained nonlinear leasts squares problem:

$$\min_{\substack{B^T B = I_{n_1} \\ C^T C = I_{n_2}}} \| A - B \otimes C \|_F .$$

A reasonable initial guess (B_0, C_0) in this setting is to set B_0 and C_0 to be the closest orthogonal matrices to the B and C that minimize $\phi_A(B,C)$.

6 Kronecker product preconditioners

To acquire some intuition about the use of Kronecker products as pre-conditioners, consider the $Ax = b$ problem where

$$A = a_1(I_{n_1} \otimes I_{n_2}) + a_2(I_{n_1} \otimes J_{n_2}) + a_2(J_{n_1} \otimes I_{n_2}) + a_3(J_{n_1} \otimes J_{n_2}), \tag{14}$$

$n = n_1 n_2$, and J_m is the m-by-m symmetric tridiagonal matrix

$$J_m = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Matrices with this structure arise in many applications. For example, the usual discretization of Poisson's equation on a rectangle with the Dirichlet stencil

$$\begin{array}{|c|c|c|}\hline a_3 & a_2 & a_3 \\ \hline a_2 & a_1 & a_2 \\ \hline a_3 & a_2 & a_3 \\ \hline \end{array} = \begin{array}{|r|r|r|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

leads to

$$A = (2I_{n_1} - J_{n_1}) \otimes I_{n_2} \; + \; I_{n_1} \otimes (2I_{n_2} - J_{n_2}). \tag{15}$$

In computer vision, the Laplace stencil defined by

$$\begin{array}{|c|c|c|}\hline a_3 & a_2 & a_3 \\ \hline a_2 & a_1 & a_2 \\ \hline a_3 & a_2 & a_3 \\ \hline \end{array} = \begin{array}{|r|r|r|}\hline -1 & -4 & -1 \\ \hline -4 & 20 & -4 \\ \hline -1 & -4 & -1 \\ \hline \end{array}$$

is frequently used and this leads to

$$A = (2I_{n_1} - J_{n_1}) \otimes (5I_{n_2} - \frac{1}{2}J_{n_2}) \; + \; (5I_{n_1} - \frac{1}{2}J_{n_1}) \otimes (2I_{n_2} - J_{n_2}).$$

In either case, if we define the constants

$$\alpha_1 = 2, \qquad \alpha_2 = 2\left(a_2 - \sqrt{a_2^2 - a_1 a_3}\right)/a_1,$$

$$\beta_1 = a_1/4, \quad \beta_2 = a_1 a_3 \left((a_2 - \sqrt{a_2^2 - a_1 a_3}\right)/4,$$

then it can be shown that

$$A = (\alpha_1 I_{n_1} + \alpha_2 J_{n_1}) \otimes (\beta_1 I_{n_2} + \beta_2 J_{n_2}) \; + \; (\beta_1 I_{n_1} + \beta_2 J_{n_1}) \otimes (\alpha_1 I_{n_2} + \alpha_2 J_{n_2}).$$

Thus, A is the sum of two Kronecker products and the remarks made in §5.5 apply. Since the rank of $\tilde{A}$ is two, the singular vectors that define the optimal B and C can be computed in $O(n)$ flops. These matrices are tridiagonal, symmetric, and positive definite in view of the discussions in §5.

Let us focus on the case when A is given by (15). For simplicity, define the 1-2-1 tridiagonal matrix

$$T_m = 2I_m - J_m$$

and note that

$$A \; = \; T_{n_1} \otimes I_{n_2} \; + \; I_{n_1} \otimes T_{n_2}.$$

From §5.5 we know that the optimizing B and C have the form

$$\begin{aligned} B &= b_1 I_{n_1} + b_2 T_{n_1} \\ C &= c_1 I_{n_2} + c_2 T_{n_2}. \end{aligned}$$

The matrix T_m has known eigenvalues:

$$Q_m^T T_m Q_m = D_m = \mathrm{diag}(\lambda_1^{(m)}, \ldots, \lambda_m^{(m)}), \qquad \lambda_j^{(m)} = 4\sin^2\left(\frac{j\pi}{2(m+1)}\right).$$

Using this result, it can be shown that the Kronecker approximation problem involves choosing b_1, b_2, c_1, and c_2 so that

$$\begin{aligned} \| A - B \otimes C \|_F^2 &= \| (T_{n_1} \otimes I_{n_2} + I_{n_1} \otimes T_{n_2}) - (b_1 I_{n_1} + b_2 T_{n_1}) \otimes (c_1 I_{n_2} + c_2 T_{n_2}) \|_F^2 \\ &= \| (D_{n_1} \otimes I_{n_2} + I_{n_1} \otimes D_{n_2}) - (b_1 I_{n_1} + b_2 D_{n_1}) \otimes (c_1 I_{n_2} + c_2 D_{n_2}) \|_F \\ &= \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left[(\lambda_i^{(n_1)} + \lambda_j^{(n_2)}) - (b_1 + b_2 \lambda_i^{(n_1)})(c_1 + c_2 \lambda_j^{(n_2)}) \right]^2 \end{aligned}$$

is minimized. The eigenvalue distribution of $M^{-1}A$, which is crucial to the success of $M = B \otimes C$ as a preconditioner, can also be examined in closed form once b_1, b_2, c_1, and c_2 are known:

$$\lambda_{ij}(M^{-1}A) = \frac{\lambda_i^{(n_1)} + \lambda_j^{(n_2)}}{(b_1 + b_2\lambda_i^{(n_1)})(c_1 + c_2\lambda_j^{(n_2)})}. \tag{16}$$

We ran some experiments in the square case $n_1 = n_2 = \sqrt{n}$. It can be shown that about $10n$ flops are required to solve a system of the form $Mz = r$ assuming that the LDL^T factorizations of B and C are available. By way of comparison, about $9n$ flops are involved when an incomplete Cholesky (IC) preconditioner is used. In the following table we compare these two preconditioners:

$\sqrt{n}$	IC Iterations	Kronecker Iterations
16	14	19
32	23	33
64	39	56
128	51	74
256	66	93

Table 1. *Comparison of Preconditioners on Model Problem*

Random right hand sides were used with termination criteria $r^T Ar \leq 10^{-6}$ where $r = b - Ax$ is the residual of the approximate solution. We have no "proof" why reasonable convergence occurs before $\sqrt{n}$ steps. A plot of the spectrum of $M^{-1}A$ using (16) reveals that many eigenvalues of $M^{-1}A$ are clustered about 1. However, the clustering is not definitive enough to suggest that $O(\sqrt{n})$ convergence is provable.

The Kronecker preconditioner applied to the above model problem compares favorably with many of the other block preconditioners that are reported in [7]. In a distributed memory environment, we suspect that the Kronecker approach may be very attractive because the preconditioner equation $CZB^T = R$ is structured perfectly for parallel computation–but that is the subject of ongoing research.

Acknowledgements

Nick Trefethen of Cornell University suggested the connection between Kronecker and rank-1 approximation. Roger Horn of the University of Utah enhanced our understanding of the rearrangement operator and pointed the way to the proof of Theorem 9.

This research was partially supported by the Cornell Theory Center, which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

References

[1] H.C. Andrews and J. Kane. Kronecker Matrices, Computer Implementation, and Generalized Spectra. *J. Assoc. Comput. Mach.*, **17**, pp 260–268, 1970.

[2] R.H. Barham and W. Drane. An Algorithm for Least Squares Estimation of Nonlinear Parameters when Some of the Parameters are Linear. *Technometrics*, **14**, pp 757–766, 1972.

[3] R.H. Bartels and G.W. Stewart. Solution of the Equation $AX + XB = C$. *Comm. ACM*, **15**, pp 820–826, 1972.

[4] J.W. Brewer. Kronecker Products and Matrix Calculus in System Theory. *IEEE Trans. on Circuits and Systems*, **25**, pp 772–781, 1978.

[5] T.F. Chan. An Optimal Circulant Preconditioner for Toeplitz Systems. *SIAM J. Sci. Stat. Comp.*, **9**, pp 766–771, 1988.

[6] R. Chan and X-Q Jin. A Family of Block Preconditioners for Block Systems. *SIAM J. Sci. Stat. Comp.*, **13**, pp 1218–1235, 1992.

[7] P. Concus, G.H. Golub, and G. Meurant. Block Preconditioning for the Conjugate Gradient Method. *SIAM J. Sci. Stat. Comp.*, **6**, pp 220–252, 1985.

[8] J. Cullum and R.A. Willoughby. *Lanczos Algorithms for Large Sparse Symmetric Eigenvalue Computations, Volume I (Theory) and II (Programs)*, Birkhauser, Boston, 1985.

[9] C. de Boor. Efficient Computer Manipulation of Tensor Products. *ACM Trans. Math. Software*, **5**, pp 173–182, 1979.

[10] D.W. Fausett and C. Fulton. Large Least Squares Problems Involving Kronecker Products. *SIAM J. Matrix Analysis*, to appear, 1992.

[11] G.H. Golub. Some Modified Eigenvalue Problems. *SIAM Review*, **15**, pp 318-344, 1973.

[12] G.H. Golub, F. Luk, and M. Overton. A Block Lanzcos Method for Computing the Singular Values and Corresponding Singular Vectors of a Matrix. *ACM Trans. Math. Soft.*, **7**, pp 149–169, 1981.

[13] G.H. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur Method for the Matrix Problem $AX + XB = C$. *IEEE Trans. Auto. Cont.*, **AC-24**, pp 909-913, 1979.

[14] G.H. Golub and V. Pereya. The Differentiation of PseudoInverses and Nonlinear least Squares Problems Whose Variables Separate. *SIAM J. Numer. Analysis*, **10**, pp 413–432, 1973.

[15] G.H. Golub and C. Van Loan. *Matrix Computations, 2nd Ed.*, Johns Hopkins University Press, Baltimore, MD, 1989.

[16] A. Graham. *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood Ltd., Chichester, England, 1981.

[17] S.R Heap and D.J. Lindler. Block Iterative Restoration of Astronomical Images with the Massively Parallel Processor. *Proc. of the First Aerospace Symposium on Massively Parallel Scientific Computation*, pp 99-109, 1986.

[18] H.V. Henderson, F. Pukelsheim, and S.R. Searle. On the History of the Kronecker Product. *Linear and Multilinear Algebra*, **14**, pp 113–120, 1983.

[19] H.V. Henderson and S.R. Searle. The Vec-Permutation Matrix, The Vec Operator and Kronecker Products: A Review. *Linear and Multilinear Algebra*, **9**, pp 271–288, 1981.

[20] R.A. Horn and C.A. Johnson. *Matrix Analysis*, Cambridge University Press, New York, 1985.

[21] R.A. Horn and C.A. Johnson. *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[22] C-H Huang, J.R. Johnson, and R.W. Johnson. Multilinear Algebra and Parallel Programming. *J. Supercomputing*, **5**, pp 189–217, 1991.

[23] J. Johnson, R.W. Johnson, D. Rodriguez, and R. Tolimieri. A Methodology for Designing, Modifying, and Implementing Fourier Transform Algorithms on Various Architectures. *Circuits, Systems, and Signal Processing*, **9**, pp 449–500, 1990.

[24] L. Kaufman. A Variable Projection Method for Solving Separable Nonlinear Least Squares Problems. *BIT*, **15**, pp 49–57, 1975.

[25] V. Pereyra and G. Scherer. Efficient Computer Manipulation of Tensor Products with Applications to Multidimensional Approximation. *Mathematics of Computation*, **27**, pp 595–604, 1973.

[26] U.A. Rauhala. Introduction to Array Algebra. *Photogrammetric Engineering and Remote Sensing*, **46**(2), pp 177–182, 1980.

[27] P.A. Regalia and S. Mitra. Kronecker Products, Unitary Matrices, and Signal Processing Applications. *SIAM Rev.*, **31**, pp 586–613, 1989.

[28] A. Swami and J. Mendel. Time and Lag Recursive Computation of Cumulants from a State-Space Model. *IEEE Trans. Auto. Cont.*, **35**, pp 4–17, 1990.

[29] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform*, SIAM Publications, Philadelphia, PA, 1992.

[30] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, New York, 1965.