SIAM J. MATRIX ANAL. APPL.
Vol. 36, No. 3, pp. 974–993

APPROXIMATING MATRICES WITH MULTIPLE SYMMETRIES*

CHARLES F. VAN LOAN† AND JOSEPH P. VOKT†

Abstract. If a tensor with various symmetries is properly unfolded, then the resulting matrix inherits those symmetries. As tensor computations become increasingly important it is imperative that we develop efficient structure preserving methods for matrices with multiple symmetries. In this paper we consider how to exploit and preserve structure in the pivoted Cholesky factorization when approximating a matrix A that is both symmetric ($A = A^T$) and what we call *perfect shuffle symmetric*, or *perf-symmetric*. The latter property means that $A = \Pi A \Pi$, where Π is a permutation with the property that $\Pi v = v$ if v is the vec of a symmetric matrix and $\Pi v = -v$ if v is the vec of a skew-symmetric matrix. Matrices with this structure can arise when an order-4 tensor $\mathcal{A}$ is unfolded and its elements satisfy $\mathcal{A}(i_1,i_2,i_3,i_4) = \mathcal{A}(i_2,i_1,i_3,i_4) = \mathcal{A}(i_1,i_2,i_4,i_3) = \mathcal{A}(i_3,i_4,i_1,i_2)$. This is the case in certain quantum chemistry applications where the tensor entries are electronic repulsion integrals. Our technique involves a closed-form block diagonalization followed by one or two half-sized pivoted Cholesky factorizations. This framework allows for a lazy evaluation feature that is important if the entries in $\mathcal{A}$ are expensive to compute. In addition to being a structure preserving rank reduction technique, we find that this approach for obtaining the Cholesky factorization reduces the work by up to a factor of 4 (see [G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, MD, 2013]).

Key words. tensor, symmetry, multilinear product, low-rank approximation

AMS subject classifications. 15A18, 15A69, 65F15

DOI. 10.1137/140986347

1. Introduction.

Low-rank approximation and the exploitation of structure are important themes throughout matrix computations. This paper revolves around some basic tensor calculations that reinforce this point. The tensors involved have multiple symmetries and the same can be said of the matrices that arise if they are obtained by a suitable unfolding.

1.1. Motivation.

Our contribution is prompted by the following problem which typically arises in Hartree–Fock calculations such as MP2 energy correction schemes [15]. Suppose $\mathcal{A} \in \mathbb{R}^{n\times n\times n\times n}$ is an order-4 tensor with the property that its entries satisfy

$$\mathcal{A}(i_1,i_2,i_3,i_4) = \begin{cases} \mathcal{A}(i_2,i_1,i_3,i_4) \\ \mathcal{A}(i_1,i_2,i_4,i_3) \\ \mathcal{A}(i_3,i_4,i_1,i_2) \end{cases}. \tag{1.1}$$

We say that such a tensor is ((1,2),(3,4))-*symmetric*; see Figure 1 for an $n = 3$ example. Given $X \in \mathbb{R}^{n\times n}$ the challenge is to compute efficiently the tensor $\mathcal{B} \in \mathbb{R}^{n\times n\times n\times n}$ defined by

$$\mathcal{B}(i_1,i_2,i_3,i_4) = \sum_{j_1,j_2,j_3,j_4=1}^{n} \mathcal{A}(j_1,j_2,j_3,j_4)X(i_1,j_1)X(i_2,j_2)X(i_3,j_3)X(i_4,j_4). \tag{1.2}$$

This is a highly structured *multilinear product*. As with many tensor computations,

*Received by the editors September 11, 2014; accepted for publication (in revised form) by M. E. Kilmer April 15, 2015; published electronically July 16, 2015. The research of the authors was partially supported by the Ronald E. McNair Post-baccalaureate Scholar program.
http://www.siam.org/journals/simax/36-3/98634.html
†Department of Computer Science, Cornell University, Ithaca, NY 14853 (cv@cs.cornell.edu, jpv52@cornell.edu).

Value	Entries that Share that Value							
1	(1,1,1,1)							
2	(2,1,1,1)	(1,2,1,1)	(1,1,2,1)	(1,1,1,2)				
3	(3,1,1,1)	(1,3,1,1)	(1,1,3,1)	(1,1,1,3)				
4	(2,2,1,1)	(1,1,2,2)						
5	(3,2,1,1)	(2,3,1,1)	(1,1,3,2)	(1,1,2,3)				
6	(3,3,1,1)	(1,1,3,3)						
7	(2,1,2,1)	(1,2,2,1)	(2,1,1,2)	(1,2,1,2)				
8	(1,3,2,1)	(3,1,2,1)	(1,3,1,2)	(3,1,1,2)	(2,1,1,3)	(1,2,1,3)	(2,1,3,1)	(1,2,3,1)
9	(2,2,2,1)	(2,2,1,2)	(2,1,2,2)	(1,2,2,2)				
10	(3,2,2,1)	(2,3,2,1)	(3,2,1,2)	(2,3,1,2)	(2,1,3,2)	(1,2,3,2)	(2,1,2,3)	(1,2,2,3)
11	(3,3,2,1)	(3,3,1,2)	(2,1,3,3)	(1,2,3,3)				
12	(3,1,3,1)	(1,3,3,1)	(3,1,1,3)	(1,3,1,3)				
13	(2,2,3,1)	(2,2,1,3)	(3,1,2,2)	(1,3,2,2)				
14	(3,2,3,1)	(2,3,3,1)	(3,2,1,3)	(2,3,1,3)	(3,1,3,2)	(1,3,3,2)	(3,1,2,3)	(1,3,2,3)
15	(3,3,3,2)	(3,3,2,3)	(3,2,3,3)	(2,3,3,3)				
16	(2,2,2,2)							
17	(3,2,2,2)	(2,3,2,2)	(2,2,3,2)	(2,2,2,3)				
18	(3,3,2,2)	(2,2,3,3)						
19	(3,2,3,2)	(2,3,3,2)	(3,2,2,3)	(2,3,2,3)				
20	(3,2,3,3)	(2,3,3,3)	(3,3,3,2)	(3,3,2,3)				
21	(3,3,3,3)							

FIG. 1. *An example of a* ((1,2),(3,4))*-symmetric tensor* ($n = 3$). *It has at most* 21 *distinct values. Equations* (1.5) *and* (1.7) *show what this tensor looks like when unfolded into a* 9×9 *matrix. In general, the subspace of* $\mathbb{R}^{n\times n\times n\times n}$ *defined by all* ((1,2),(3,4))*-symmetric tensors has dimension* $(n^4 + 2n^3 + 3n^2 + 2n)/8$.

(1.2) can be reformulated as a matrix computation. In particular, it can be shown that

$$B = (X \otimes X)A(X \otimes X)^T, \tag{1.3}$$

where A and B are n^2-by-n^2 matrices that are obtained by certain unfoldings of the tensors $\mathcal{A}$ and $\mathcal{B}$. Depending upon the chosen unfolding, the matrices A and B inherit the tensor symmetries (1.1).

For example, suppose $A = \mathcal{A}_{[1,3]\times[2,4]}$ is the "$[1,3] \times [2,4]$ unfolding" defined by

$$\mathcal{A}(i_1, i_2, i_3, i_4) \to A(i_1 + (i_3 - 1)n, i_2 + (i_4 - 1)n). \tag{1.4}$$

This n^2-by-n^2 matrix can be regarded as an n-by-n block matrix $A = (A_{pq})$ whose blocks A_{pq} are n-by-n matrices. It follows from (1.4) that

$$\mathcal{A}(i_1, i_2, i_3, i_4) = [A_{i_3,i_4}]_{i_1,i_2}.$$

Combining this with (1.1) we conclude that $A_{qp} = A_{pq} = A_{pq}^T$. Note that this implies $A^T = A$. To visualize the structure associated with the $[1,3]\times[2,4]$ unfolding, suppose that $\mathcal{A} \in \mathbb{R}^{3\times3\times3\times3}$ is defined by Figure 1. It follows that

$$\mathcal{A}_{[1,3]\times[2,4]} = \left[\begin{array}{ccc|ccc|ccc} 1 & 2 & 3 & 2 & 7 & 8 & 3 & 8 & 12 \\ 2 & 4 & 5 & 17 & 19 & 10 & 8 & 13 & 14 \\ 3 & 5 & 6 & 8 & 10 & 11 & 12 & 14 & 15 \\ \hline 2 & 7 & 8 & 4 & 9 & 13 & 5 & 10 & 14 \\ 7 & 9 & 10 & 9 & 16 & 17 & 10 & 17 & 19 \\ 8 & 10 & 11 & 13 & 17 & 18 & 14 & 19 & 20 \\ \hline 3 & 8 & 12 & 5 & 10 & 14 & 6 & 11 & 15 \\ 8 & 13 & 14 & 10 & 17 & 19 & 11 & 18 & 20 \\ 12 & 14 & 15 & 14 & 19 & 20 & 15 & 20 & 21 \end{array}\right]. \tag{1.5}$$

On the other hand, the $[\,1,2\,]\times[\,3,4\,]$ unfolding $A=\mathcal{A}_{[1,2]\times[3,4]}$ defined by

$$\mathcal{A}(i_1,i_2,i_3,i_4)\rightarrow A(i_1+(i_2-1)n,i_3+(i_4-1)n) \tag{1.6}$$

results in a matrix A with different properties. Indeed, if we apply this mapping to the tensor defined in Figure 1, then we obtain

$$\mathcal{A}_{[1,2]\times[3,4]}=\left[\begin{array}{ccc|ccc|ccc}
1&2&3&2&4&5&3&5&6\\
2&7&8&7&9&10&8&10&11\\
3&8&12&8&13&14&12&14&15\\
\hline
2&7&8&7&9&10&8&10&11\\
4&9&13&9&16&17&13&17&18\\
5&10&14&10&17&19&14&19&20\\
\hline
3&8&12&8&13&14&12&14&15\\
5&10&14&10&17&19&14&19&20\\
6&11&15&11&18&20&15&20&21
\end{array}\right]. \tag{1.7}$$

It is easy to prove that this unfolding is also symmetric. (Just combine (1.6) with the observation that $\mathcal{A}(i_1,i_2,i_3,i_4)=\mathcal{A}(i_3,i_4,i_1,i_2)$.) But it also satisfies a type of symmetry that is related to a particular perfect shuffle permutation. To see this we define the n^2-by-n^2 permutation matrix Π_{nn} by

$$\Pi_{nn}=I_{n^2}(:,p),\qquad p=[\,1{:}n{:}n^2\mid 2{:}n{:}n^2\mid\cdots\mid n{:}n{:}n^2\,], \tag{1.8}$$

where we are making use of the MATLAB colon notation. Here is an example:

$$\Pi_{33}=\left[\begin{array}{ccc|ccc|ccc}
1&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0\\
\hline
0&1&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&1&0\\
\hline
0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&1
\end{array}\right]=I_9(:,[\,1\,4\,7\,2\,5\,8\,3\,6\,9\,]). \tag{1.9}$$

A matrix $A\in\mathbb{R}^{n^2\times n^2}$ is *perfect shuffle invariant* if

$$A=\Pi_{nn}A\Pi_{nn} \tag{1.10}$$

and *PS-symmetric* if it is both symmetric and perfect shuffle invariant.

In section 2 we show how to construct a reduced rank approximation to a PS-symmetric matrix that is also PS-symmetric. This is important in the evaluation of the multilinear product (1.2). In particular, it enables us to approximate the unfolding matrix A in (1.4) with a relatively short sum of structured Kronecker products:

$$A\approx\sum_{i=1}^{r}\sigma_i\cdot C_i\otimes C_i\qquad C_i^T=C_i\in\mathbb{R}^{n\times n},\ r\ll n^2. \tag{1.11}$$

It then follows from (1.3) that

$$B \approx \sum_{i=1}^{r} \sigma_i \cdot (X \otimes X)(C_i \otimes C_i)(X \otimes X)^T = \sum_{i=1}^{r} \sigma_i (X C_i X^T) \otimes (X C_i X^T). \tag{1.12}$$

The $\{\sigma_i, C_i, X\}$ representation of B (and hence $\mathcal{B}$) is an $O(rn^3)$ computation. There are r steps of symmetric matrix times matrix ($C_i \times X^T$) followed by matrix times matrix $X \times (C_i \times X^T)$. Assuming matrix multiply takes cn^3 flops, without using the fact that the C_i are symmetric, it would take $r(cn^3 + cn^3) = 2crn^3$. Assuming that symmetric matrix times matrix takes $cn^3/2$ flops, and using the symmetry of C_i, it would take $r(cn^3/2 + cn^3) = (3/2)crn^3$. So we can reduce work in computing the factorization of B given the factorization of A by a factor of $2crn^3/((3/2)crn^3) = 4/3$ by using the symmetry of C_i.

The expansion (1.11) looks like a Kronecker-product SVD of A [6, 19, 18]. However, the method that we propose in this paper is not based on expensive SVD-like computations but on a structured factorization that combines block diagonalization with a pair of "half-size" pivoted Cholesky factorizations. Recall that if $M \in \mathbb{R}^{N \times N}$ is symmetric and positive semidefinite with $r \leq N$ positive eigenvalues, then the pivoted Cholesky factorization (in exact arithmetic) computes the factorization

$$PMP^T = LL^T,$$

where $P \in \mathbb{R}^{N \times N}$ is a permutation matrix and $L \in \mathbb{R}^{N \times r}$ is lower triangular with $r = \text{rank}(A)$. It follows that if $Y = P^T L = [y_1, \ldots, y_r]$ is a column partitioning, then

$$M = (P^T L)(P^T L)^T = YY^T = \sum_{k=1}^{r} y_k y_k^T.$$

In practice, r is (numerically) determined during the factorization process. More on this in section 4.2. We mention that this rank-r representation requires $Nr^2 - 2r^2/3 + O(Nr)$ flops; see [6, pp. 165–166] for more details.[1]

1.2. Overview of the paper. In section 2 we discuss the properties of PS-symmetric matrices. A key result is the derivation of a simple orthogonal matrix Q that can be used to block diagonalize a PS-symmetric matrix A: $Q^T AQ = \text{diag}(A_1, A_2)$. Rank-revealing pivoted Cholesky factorizations are then applied to the half-sized diagonal blocks. The resulting factor matrices are then combined with Q to produce a rank-1 expansion for A with terms that are also PS-symmetric. In section 3 we apply these results to compute a structured multilinear product whose defining tensor $\mathcal{A}$ is ((1,2),(3,4))-symmetric. An application from quantum chemistry is considered that has a dramatic low-rank feature. Implementation details and benchmarks are provided in section 4. Anticipated future work and conclusions are offered in section 5.

[1] The connection between the pivoted LDL factorization $PMP^T = \tilde{L}D\tilde{L}^T$, where $\tilde{L}$ is unit lower triangular and the pivoted Cholesky factorization $PMP^T = LL^T$ is simple. The lower triangular Cholesky factor L is given by $L = \tilde{L} \cdot \text{diag}(d_1^{1/2}, \ldots, d_r^{1/2})$. Virtually all of the rank-revealing operations in this paper can be framed in "LDL" language. We use the Cholesky representation so that readers can more easily relate our work to what has already been published and to existing procedures in LAPACK.

1.3. Centrosymmetry: An instructive preview. We conclude the introduction with a brief discussion of matrices that are centrosymmetric. These are matrices that are symmetric about their diagonal *and* antidiagonal, e.g.,

$$A = \begin{bmatrix} a & b & c & d \\ b & e & f & c \\ c & f & e & b \\ d & c & b & a \end{bmatrix}.$$

They are a particularly simple class of multisymmetric matrices and because of that they can be used to anticipate the main ideas that follow in sections 2 and 3. For a more in-depth treatment of centrosymmetry, see Andrew [1], Datta and Morgera [3], and Pressman [10].

Formally, a matrix $A \in \mathbb{R}^{n\times n}$ is *centrosymmetric* if $A = A^T$ and $A = E_n A E_n$, where

$$E_n = I_n(:, n{:}{-}1{:}1) \in \mathbb{R}^{n\times n}$$

is the n-by-n *exchange permutation*. The redundancies among the elements of a centrosymmetric matrix are nicely exposed through blocking. Assume for clarity that $n = 2m$. (The odd-n case is basically the same.) If

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad A_{ij} \in \mathbb{R}^{m\times m},$$

is centrosymmetric, then by substituting

$$E_n = \begin{bmatrix} 0 & E_m \\ E_m & 0 \end{bmatrix}$$

into the equation $A = E_n A E_n$ we see that $A_{21} = E_m A_{12} E_m$ and $A_{22} = E_m A_{11} E_m$, i.e.,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ E_m A_{12} E_m & E_m A_{11} E_m \end{bmatrix}. \tag{1.13}$$

Moreover, A_{11} and $A_{12}E_m$ are each symmetric. Given this block structure it is easy to confirm that the orthogonal matrix

$$Q_{\scriptscriptstyle E} = \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} I_m & I_m \\ E_m & -E_m \end{array}\right] \equiv \left[\begin{array}{c|c} Q_+ & Q_- \end{array}\right] \tag{1.14}$$

block diagonalizes A:

$$Q_{\scriptscriptstyle E}^T A Q_{\scriptscriptstyle E} = \begin{bmatrix} A_{11} + A_{12}E_m & 0 \\ 0 & A_{11} - A_{12}E_m \end{bmatrix} \equiv \begin{bmatrix} A_+ & 0 \\ 0 & A_- \end{bmatrix}. \tag{1.15}$$

If A is positive semidefinite, then the same can be said of A_+ and A_- and we can compute the following half-sized pivoted Cholesky factorizations,

$$P_+ A_+ P_+^T = L_+ L_+^T, \tag{1.16}$$

$$P_- A_- P_-^T = L_- L_-^T. \tag{1.17}$$

TABLE 1

T_u is the time required to compute the Cholesky factorization of A (the unstructured algorithm). T_s is the time required to set up A_+ and A_- and compute their Cholesky factorizations (the structured algorithm). $T_{\text{set-up}}$ is the time required to just set-up A_+ and A_-. The LAPACK procedures POTRF (unpivoted Cholesky calling level-3 BLAS) and PSTRF (pivoted Cholesky calling level-3 BLAS) were used for full-rank and low-rank cases, respectively. Results are based on running numerous random trials for each combination of n and (r_+, r_-). A single core of the Intel(R) Core(TM) i5-3210M CPU @ 2.50 GHz was used.

	$r_+ = n/2\ ,\ r_- = n/2$		$r_+ = n/100\ ,\ r_- = n/100$	
n	T_u/T_s	$T_{set\text{-}up}/T_s$	T_u/T_s	$T_{set\text{-}up}/T_s$
1500	1.93	0.32	0.53	0.66
3000	2.68	0.22	0.56	0.69
4500	2.89	0.17	0.83	0.64
6000	3.18	0.13	0.88	0.65

If we define the matrices $Y_+ \in \mathbb{R}^{n\times m}$ and $Y_- \in \mathbb{R}^{n\times m}$ by

$$Y_+ = Q_+ P_+^T L_+ = [\, y_+^{(1)} \mid \cdots \mid y_+^{(m)} \,], \tag{1.18}$$

$$Y_- = Q_- P_-^T L_- = [\, y_-^{(1)} \mid \cdots \mid y_-^{(m)} \,], \tag{1.19}$$

then it follows from (1.15)–(1.19) that

$$A = Y_+ Y_+^T + Y_- Y_-^T = \sum_{i=1}^{m} y_+^{(i)} [y_+^{(i)}]^T + \sum_{i=1}^{m} y_-^{(i)} [y_-^{(i)}]^T.$$

Each of the rank-1 matrices in this expansion is centrosymmetric because $E_n Y_+ = Y_+$ and $E_n Y_- = -Y_-$. It follows that if $r_+ \le m$ and $r_- \le m$, then

$$A_{\{r_+,r_-\}} = \sum_{i=1}^{r_+} y_+^{(i)} [y_+^{(i)}]^T + \sum_{i=1}^{r_-} y_-^{(i)} [y_-^{(i)}]^T \tag{1.20}$$

is centrosymmetric and $\operatorname{rank}(A_{\{r_+,r_-\}}) = r_+ + r_-$. Thus, by combining the block diagonalization (1.15) with the pivoted Cholesky factorizations (1.16)–(1.17) we can approximate a given positive semidefinite centrosymmetric matrix with a matrix of lower rank that is also centrosymmetric.

We briefly consider the efficiency of such a maneuver keeping in mind the "preview nature" of this subsection. Here are some obvious implementation concerns:

1. What is the cost of the block diagonalization? The matrices A_+ and A_- are simple enough, but is their formation a negligible overhead?
2. From the flop point of view, halving the dimension of an $O(n^3)$ factorization reduces the volume of arithmetic by a factor of 8. Is the cost of computing the pivoted Cholesky's of A_+ and A_- one-fourth the cost of the single full-size pivoted Cholesky of A?
3. If A is close to a matrix with very low rank and/or its entries a_{ij} are expensive to compute, then it may be preferable to work with a left-looking implementation of pivoted Cholesky that computes matrix entries on a "need to know" basis. How can one organize pivot determination in such a setting?

Table 1 sheds light on some of these issues by comparing the computation of the structured approximation $A_{\{r_+,r_-\}}$ with the unstructured approximation A_r based on

$PAP^T = LL^T$, i.e.,

$$A_r = \sum_{i=1}^{r} y^{(i)}[y^{(i)}]^T,$$

where $P^T L = [y^{(1)}, \ldots, y^{(r)}]$ and $r = r_+ + r_-$.

In the full rank case ($r_+ = n/2\,,\ r_- = n/2$), a flop-only analysis would predict a speed-up factor of 4 since we are replacing one n-by-n Cholesky factorization ($n^3/3$ flops) with a pair of 2 half-size factorizations ($2 \cdot (n/2)^3/3$ flops.) The ratios T_u/T_s are somewhat less than this because there is an $O(n^2)$ overhead associated with the setting up of the matrices A_+ and A_-. This is quantified by the ratios $T_{\text{set-up}}/T_s$.

The low-rank results point to the importance of having a "lazy evaluation" strategy when it comes to setting up the matrices A_+ and A_-. The LAPACK routines that are used are right looking and thus require the complete $O(n^2)$ set-up of these half-sized matrices. However, the flop cost of the pivoted Cholesky factorizations of these low-rank matrices is $O(nr^2)$. Thus, the set-up costs dominate and there is a serious tension between efficiency and structure preservation. What we need is a *left-looking* pivoted Cholesky procedure that involves an $O(nr)$ set-up cost. We shall discuss just such a framework in section 4.2 in the context of a highly structured low-rank PS-symmetric approximation problem.

2. Perfect shuffle symmetry. Just as centrosymmetry is defined by the equation $A = E_n A E_n$, PS-symmetry is defined by the equation $A = \Pi_{nn} A \Pi_{nn}$, where Π_{nn} is a particular perfect shuffle permutation. We start by looking at the eigenvectors of this permutation. This leads to the construction of a simple orthogonal matrix (like Q_E in (1.14)) that can be used to block diagonalize a PS-symmetric matrix. A framework for structured low-rank approximation follows.

2.1. Perfect shuffle properties. Perfect shuffle permutations relate matrix transposition to vector permutation. Following Van Loan [17, p. 78], if $m = pr$, then the perfect shuffle permutation $\Pi_{pr} \in \mathbb{R}^{m \times m}$ is defined by

$$\Pi_{pr} = I_n(:, [(1{:}r{:}m)\ (2{:}r{:}m) \cdots (r{:}r{:}m)]).$$

The action of Π_{pr} is best described using the MATLAB `reshape` operator, e.g.,

$$x \in \mathbb{R}^{12} \Rightarrow \texttt{reshape}(x, 3, 4) = \begin{bmatrix} x_1 & x_4 & x_7 & x_{10} \\ x_2 & x_5 & x_8 & x_{11} \\ x_3 & x_6 & x_9 & x_{12} \end{bmatrix}.$$

If $x \in \mathbb{R}^{pr}$, then

$$y = \Pi_{pr} x \Rightarrow \texttt{reshape}(y, p, r) = \texttt{reshape}(x, r, p)^T. \tag{2.1}$$

In other words, if $S \in \mathbb{R}^{r \times p}$, then $\texttt{vec}(S^T) = \Pi_{pr}\texttt{vec}(S)$.

We shall be interested in the case $p = r = n$. Using (2.1) it is easy to see that $\Pi_{nn}\Pi_{nn} = I$ showing that $\Pi_{nn} = \Pi_{nn}^T$. Thus, if λ is an eigenvalue of Π_{nn}, then $\lambda = 1$ or $\lambda = -1$. Using (2.1) again it follows that

$$\Pi_{nn} x = +x \Rightarrow S = \texttt{reshape}(x, n, n) \text{ is symmetric,}$$
$$\Pi_{nn} x = -x \Rightarrow S = \texttt{reshape}(x, n, n) \text{ is skew-symmetric.}$$

Thus, $\Pi_{nn} x = x$ if and only if $S = S^T$. Likewise, $\Pi_{nn} x = -x$ if and only if $S = -S^T$.

Using these observations about Π_{nn}, it is easy to verify that the entries in a PS-symmetric matrix $A = \Pi_{nn} A \Pi_{nn}$ satisfy

$$\begin{aligned} & A(i_1 + (i_2-1)n, j_1 + (j_2-1)n) \\ = {} & A(j_1 + (j_2-1)n, i_1 + (i_2-1)n) \\ = {} & A(i_2 + (i_1-1)n, j_2 + (j_1-1)n) \\ = {} & A(j_2 + (j_1-1)n, i_2 + (i_1-1)n), \end{aligned} \tag{2.2}$$

where it is understood that the indices i_1, i_2, j_1, and j_2 range from 1 to n.

2.2. Block diagonalization. Define the subspaces $\mathbf{S}_{nn}^{(\mathrm{sym})}$ and $\mathbf{S}_{nn}^{(\mathrm{skew})}$ by

$$\mathbf{S}_{nn}^{(\mathrm{sym})} = \{ x \in \mathbb{R}^{n^2} \mid \Pi_{nn} x = x \}, \tag{2.3}$$

$$\mathbf{S}_{nn}^{(\mathrm{skew})} = \{ x \in \mathbb{R}^{n^2} \mid \Pi_{nn} x = -x \}. \tag{2.4}$$

It is easy to verify that $[\mathbf{S}_{nn}^{(\mathrm{sym})}]^{\perp} = \mathbf{S}_{nn}^{(\mathrm{skew})}$. Moreover, if $A \in \mathbb{R}^{n^2 \times n^2}$ is PS-symmetric and $x \in \mathbf{S}_{nn}^{(\mathrm{sym})}$, then

$$Ax = (\Pi_{nn} A \Pi_{nn}) x = (\Pi_{nn} A)(\Pi_{nn} x) = (\Pi_{nn} A) x = \Pi_{nn}(Ax)$$

which shows that $\mathbf{S}_{nn}^{(\mathrm{sym})}$ is an invariant subspace for A. The subspace $\mathbf{S}_{nn}^{(\mathrm{skew})}$ is also invariant for A by similar reasoning.

Using these facts we can construct a sparse orthogonal matrix Q_{nn} that can be used to block diagonalize a PS-symmetric matrix. Let $I_n = [e_1, \ldots, e_n]$ be a column partitioning and define the matrices

$$\begin{aligned} Q_{nn}^{(\mathrm{sym})} &= \left[\begin{array}{c|c|c} q_1^{(\mathrm{sym})} & \cdots & q_{n_{\mathrm{sym}}}^{(\mathrm{sym})} \end{array} \right], \qquad n_{\mathrm{sym}} = n(n+1)/2, \\ Q_{nn}^{(\mathrm{skew})} &= \left[\begin{array}{c|c|c} q_1^{(\mathrm{skew})} & \cdots & q_{n_{\mathrm{skew}}}^{(\mathrm{skew})} \end{array} \right], \qquad n_{\mathrm{skew}} = n(n-1)/2, \end{aligned} \tag{2.5}$$

as follows:

```
k = 0
for j = 1:n
    for i = j:n
        k = k + 1
        q_k^(sym) = { (e_i⊗e_j + e_j⊗e_i)/√2   if i > j
                    { e_i⊗e_i                  if i = j
    end
end
k = 0
for j = 1:n − 1
    for i = j + 1:n
        k = k + 1
        q_k^(skew) = (e_j⊗e_i − e_i⊗e_j)/√2
    end
end
```

(2.6)

Since $\texttt{reshape}(e_i \otimes e_j, n, n) = e_j e_i^T$, it is clear that the columns of $Q^{(\mathrm{sym})}$ reshape to symmetric matrices while the columns of $Q^{(\mathrm{skew})}$ reshape to skew-symmetric matrices. Define the n^2-by-n^2 matrix

$$Q_{nn} = \left[Q_{nn}^{(\mathrm{sym})} \mid Q_{nn}^{(\mathrm{skew})} \right], \tag{2.7}$$

e.g.,

$$Q_{33} = \left[\begin{array}{cccccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & \alpha \\ 0 & 0 & \alpha & 0 & 0 & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & -\alpha \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right], \qquad \alpha = \frac{1}{\sqrt{2}}.$$

It is clear that this matrix is orthogonal. Here is a formal proof together with a verification that Q_{nn} block diagonalizes a matrix with PS-symmetry.

THEOREM 2.1. *If $A \in \mathbb{R}^{n^2 \times n^2}$ is PS-symmetric and Q_{nn} is defined by* (2.6), *then Q_{nn} is orthogonal and*

$$Q_{nn}^T A Q_{nn} = \begin{bmatrix} A^{(\mathrm{sym})} & 0 \\ 0 & A^{(\mathrm{skew})} \end{bmatrix}, \tag{2.8}$$

where $A^{(\mathrm{sym})} \in \mathbb{R}^{n_{\mathrm{sym}} \times n_{\mathrm{sym}}}$ and $A^{(\mathrm{skew})} \in \mathbb{R}^{n_{\mathrm{skew}} \times n_{\mathrm{skew}}}$.

Proof. All the columns in Q_{nn} have unit length so the problem is to establish that any pair of its columns are orthogonal to each other. It is obvious that $\{e_1 \otimes e_1, \ldots, e_n \otimes e_n\}$ is an orthonormal set of vectors and that

$$(e_i \otimes e_i)^T (e_p \otimes e_q) = (e_i^T e_p)(e_i^T e_q) = 0$$

provided $p \neq q$. It follows that any column of the form $e_i \otimes e_i$ is orthogonal to all the other columns in Q_{nn}. Using the Kronecker delta δ_{ij}, if $i \neq j$ and $p \neq q$, then

$$(e_i \otimes e_j + e_j \otimes e_i)^T)(e_p \otimes e_q - e_q \otimes e_p) = \delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp} - \delta_{iq}\delta_{jp} - \delta_{jq}\delta_{ip} = 0.$$

This confirms that

$$\left[Q_{nn}^{(\mathrm{skew})} \right]^T Q_{nn}^{(\mathrm{sym})} = 0. \tag{2.9}$$

If (i,j), (j,i), (p,r), and (r,p) are distinct index pairs, then it is easy to show that

$$\begin{aligned} (e_i \otimes e_j + e_j \otimes e_i)^T)(e_p \otimes e_q + e_q \otimes e_p) &= \delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp} + \delta_{iq}\delta_{jp} + \delta_{jq}\delta_{ip} = 0, \\ (e_i \otimes e_j - e_j \otimes e_i)^T)(e_p \otimes e_q - e_q \otimes e_p) &= \delta_{ip}\delta_{jq} - \delta_{iq}\delta_{jp} - \delta_{iq}\delta_{jp} + \delta_{jq}\delta_{ip} = 0. \end{aligned}$$

These equations establish that the columns of both $Q_{nn}^{(\mathrm{sym})}$ and $Q_{nn}^{(\mathrm{skew})}$ are orthonormal. Combined with (2.9) we see that Q_{nn} is an orthogonal matrix.

To confirm that this matrix block diagonalizes a PS-symmetric A we observe using (2.7) that

$$Q_{nn}^T A Q_{nn} = \begin{bmatrix} \left[Q_{nn}^{(\mathrm{sym})}\right]^T A\, Q_{nn}^{(\mathrm{sym})} & \left[Q_{nn}^{(\mathrm{sym})}\right]^T A\, Q_{nn}^{(\mathrm{skew})} \\ \left[Q_{nn}^{(\mathrm{skew})}\right]^T A\, Q_{nn}^{(\mathrm{sym})} & \left[Q_{nn}^{(\mathrm{skew})}\right]^T A\, Q_{nn}^{(\mathrm{skew})} \end{bmatrix}.$$

Since $\Pi_{nn} Q_{nn}^{(\mathrm{sym})} = Q_{nn}^{(\mathrm{sym})}$ and $\Pi_{nn} Q_{nn}^{(\mathrm{skew})} = -Q_{nn}^{(\mathrm{skew})}$ it follows that

$$[Q_{nn}^{(\mathrm{sym})}]^T A Q_{nn}^{(\mathrm{skew})} = [Q_{nn}^{(\mathrm{sym})}]^T \Pi_{nn} A \Pi_{nn} Q_{nn}^{(\mathrm{skew})} = -[Q_{nn}^{(\mathrm{sym})}]^T A Q_{nn}^{(\mathrm{skew})} = 0.$$

Setting

$$A^{(\mathrm{sym})} = Q_{nn}^{(\mathrm{sym})T}\, A\, Q_{nn}^{(\mathrm{sym})}, \tag{2.10}$$

$$A^{(\mathrm{skew})} = Q_{nn}^{(\mathrm{skew})T}\, A\, Q_{nn}^{(\mathrm{skew})}, \tag{2.11}$$

completes the proof of the theorem. ∎

The efficient formation of $A^{(\mathrm{sym})}$ and $A^{(\mathrm{skew})}$ is critical to our method and to that end we develop characterization of these blocks that is much more useful than (2.10) and (2.11). Define the index vectors $\mathtt{sym_n} \in \mathbb{R}^{n_{\mathrm{sym}}}$ and $\mathtt{skew_n} \in \mathbb{R}^{n_{\mathrm{skew}}}$ as follows:

```
k = 0
for j = 1:n
    for i = j:n
        k = k + 1
        sym_n(k) = i + (j − 1)n
    end
end
```

```
k = 0
for j = 1:n
    for i = j + 1:n
        k = k + 1
        skew_n(k) = i + (j − 1)n
    end
end
```

(2.12)

If $M \in \mathbb{R}^{n\times n}$ and $v = \mathtt{vec}(M)$, then $v(\mathtt{sym}_n)$ is the vector of M's lower triangular entries and $v(\mathtt{skew}_n)$ is the vector of M's strictly lower triangular entries. (Consider the example $\mathtt{sym}_3 = [1\ 2\ 3\ 5\ 6\ 9]$ and $\mathtt{skew}_3 = [2\ 3\ 6]$.) Since

$$\Pi_{nn}\,(e_i \otimes e_j) = (e_j \otimes e_i),$$

it follows from (2.6) that if

$$T^{(\mathrm{sym})} = \frac{I_{n^2} + \Pi_{nn}}{2}, \tag{2.13}$$

then $q_k^{(\mathrm{sym})}$ is a multiple of $T^{(\mathrm{sym})}(:, \mathtt{sym_n}(k))$ while if

$$T^{(\mathrm{skew})} = \frac{I_{n^2} - \Pi_{nn}}{2}, \tag{2.14}$$

then $q_k^{(\mathrm{skew})}$ is a multiple of $T^{(\mathrm{skew})}(:, \mathtt{skew_n}(k))$. Indeed, if the n^2-by-n^2 diagonal matrix $\Delta^{(\mathrm{sym})}$ is defined by

$$\Delta^{(\mathrm{sym})}_{i+(j-1)n,i+(j-1)n} = \begin{cases} \sqrt{2}, & i \neq j, \\ 1, & i = j, \end{cases} \tag{2.15}$$

where i and j each range from 1 to n, then it is easy to verify that the columns of $T^{(\mathrm{sym})}\Delta^{(\mathrm{sym})}$ have unit 2-norm and

$$Q^{(\mathrm{sym})}_{nn} = T^{(\mathrm{sym})}(:, u) \cdot \Delta^{(\mathrm{sym})}(u, u), \qquad u = \mathtt{sym_n}. \tag{2.16}$$

The scaling to obtain $Q^{(\mathrm{skew})}_{nn}$ is simpler:

$$Q^{(\mathrm{skew})}_{nn} = \sqrt{2} \cdot T^{(\mathrm{skew})}(:, v), \qquad v = \mathtt{skew_n}. \tag{2.17}$$

Note that $T^{(\mathrm{sym})}$ is symmetric and $T^{(\mathrm{sym})}T^{(\mathrm{sym})} = T^{(\mathrm{sym})}$. Since $x \in \mathbf{S}^{(\mathrm{sym})}_{nn}$ implies $T^{(\mathrm{sym})}x = x$, it follows that $T^{(\mathrm{sym})}$ is the orthogonal projector associated with $\mathbf{S}^{(\mathrm{sym})}_{nn}$. Likewise, $T^{(\mathrm{skew})}$ is the orthogonal projector associated with $\mathbf{S}^{(\mathrm{skew})}_{nn}$.

Since $\Pi_{nn}A\Pi_{nn} = A$, it is easy to show that

$$\begin{aligned} T^{(\mathrm{sym})T}\, A\, T^{(\mathrm{sym})} &= (I_{n^2} + \Pi_{nn})A(I_{n^2} + \Pi_{nn}) = (A + A\Pi_{nn})/2, \\ T^{(\mathrm{skew})T}\, A\, T^{(\mathrm{skew})} &= (I_{n^2} - \Pi_{nn})A(I_{n^2} - \Pi_{nn}) = (A - A\Pi_{nn})/2. \end{aligned}$$

When these equations are combined with (2.10), (2.11), (2.16), and (2.17) we obtain

$$\begin{aligned} A^{(\mathrm{sym})} &= \Delta^{(\mathrm{sym})}(u, u) \cdot \frac{A(u, u) + A(u, :)\Pi_{nn}(:, u)}{2} \cdot \Delta^{(\mathrm{sym})}(u, u), \\ A^{(\mathrm{skew})} &= (A(v, v) - A(v, :)\Pi_{nn}(:, v)). \end{aligned}$$

This can be rewritten as

$$A^{(\mathrm{sym})} = \Delta^{(\mathrm{sym})}(u, u) \cdot \frac{A(u, u) + A(u, p(u))}{2} \cdot \Delta^{(\mathrm{sym})}(u, u), \tag{2.18}$$

$$A^{(\mathrm{skew})} = A(v, v) - A(v, p(v)), \tag{2.19}$$

where $p = \left[\, 1{:}n{:}n^2 \;\; 2{:}n{:}n^2 \cdots n{:}n{:}n^2 \,\right]$ is the index vector that defines Π_{nn}, i.e., $\Pi_{nn} = I_{n^2}(:, p)$; see (1.18).

2.3. The Schur decomposition and SVD of a PS-symmetric matrix. It is not a surprise that the Schur decomposition of a PS-symmetric matrix involves a highly structured eigenvector matrix. If

$$[U^{(\mathrm{sym})}]^T A^{(\mathrm{sym})} U^{(\mathrm{sym})} = \Lambda^{(\mathrm{sym})} = \mathrm{diag}(\lambda_1^{(\mathrm{sym})}, \ldots, \lambda_{n_{\mathrm{sym}}}^{(\mathrm{sym})})$$

and

$$[U^{(\mathrm{skew})}]^T A^{(\mathrm{skew})} U^{(\mathrm{skew})} = \Lambda^{(\mathrm{skew})} = \mathrm{diag}(\lambda_1^{(\mathrm{skew})}, \ldots, \lambda_{n_{\mathrm{skew}}}^{(\mathrm{skew})})$$

are the Schur decompositions of the diagonal blocks in (2.8) and the orthogonal matrix Q is defined by

$$Q = Q_{nn} \begin{bmatrix} U^{(\mathrm{sym})} & 0 \\ 0 & U^{(\mathrm{skew})} \end{bmatrix} = \left[\; Q^{(\mathrm{sym})}_{nn} U^{(\mathrm{sym})} \;\middle|\; Q^{(\mathrm{skew})}_{nn} U^{(\mathrm{skew})} \;\right], \tag{2.20}$$

then

$$Q^T A Q = \begin{bmatrix} D^{(\mathrm{sym})} & 0 \\ 0 & D^{(\mathrm{skew})} \end{bmatrix} = \mathrm{diag}(\lambda_1^{(\mathrm{sym})}, \ldots, \lambda_{n_{\mathrm{sym}}}^{(\mathrm{sym})}, \lambda_1^{(\mathrm{skew})}, \ldots, \lambda_{n_{\mathrm{skew}}}^{(\mathrm{skew})}). \tag{2.21}$$

By virtue of how we defined Q_{nn} in (2.6), the columns of $Q_{nn}^{(\mathrm{sym})} U^{(\mathrm{sym})}$ (the "sym eigenvectors") reshape to n-by-n symmetric matrices. Likewise, the columns of $Q_{nn}^{(\mathrm{skew})} U^{(\mathrm{skew})}$ (the "skew eigenvectors") reshape to n-by-n skew-symmetric matrices.

Note that this structured Schur decomposition is an unnormalized SVD of A. The singular values of A are the absolute values of the λ's. Reordering together with some "minus one" scalings can turn (2.20) and (2.21) into a normalized SVD.

2.4. The Kronecker product SVD of a PS-symmetric matrix. A block matrix with uniformly sized blocks has a *Kronecker product SVD* (KPSVD); see [6, pp. 712–714]. For example, if A is an n-by-n block matrix with n-by-n blocks, then there exist n-by-n matrices $B_1, \ldots, B_{n^2}$ and $C_1, \ldots, C_{n^2}$ and scalars $\sigma_1 \geq \cdots \geq \sigma_{n^2} \geq 0$ such that

$$A = \sum_{k=1}^{n^2} \sigma^{(k)} \, (B_k \otimes C_k).$$

The decomposition is related to the SVD of the n^2-by-n^2 matrix $\tilde{A}$ defined by

$$\tilde{A}(i_2 + (j_2 - 1)n, i_1 + (j_1 - 1)n) = A(i_1 + (i_2 - 1)n, j_1 + (j_2 - 1)n), \tag{2.22}$$

where the indices i_1, i_2, j_1, and j_2 range from 1 to n. In particular, if

$$\tilde{A} = \sum_{k=1}^{n^2} \sigma_k \, b_k c_k^T$$

is the rank-1 SVD expansion of $\tilde{A}$, then

$$B_k = \mathtt{reshape}(b_k, n, n), \tag{2.23}$$

$$C_k = \mathtt{reshape}(c_k, n, n) \tag{2.24}$$

for $k = 1{:}n^2$.

We show that the KPSVD of a PS-symmetric matrix is highly structured. To begin with, the matrix $\tilde{A}$ defined by (2.22) is PS-symmetric. Indeed by combining (2.2) and (2.22) we see that

$$\begin{aligned} &\tilde{A}(i_2 + (j_2 - 1)n, i_1 + (j_1 - 1)n) \\ &= \tilde{A}(j_2 + (i_2 - 1)n, j_1 + (i_1 - 1)n) \\ &= \tilde{A}(i_1 + (j_1 - 1)n, i_2 + (j_2 - 1)n) \\ &= \tilde{A}(j_1 + (i_1 - 1)n, j_2 + (i_2 - 1)n). \end{aligned} \tag{2.25}$$

These equalities show that $\tilde{A}^T = \tilde{A}$ and $\tilde{A} = \Pi_{nn} \tilde{A} \Pi_{nn}$. In other words, $\tilde{A}$ is PS-symmetric. From (2.18) and (2.19) we know that $\tilde{A}$ has a rank-1 Schur decomposition expansion of the form

$$\tilde{A} = \sum_{i=1}^{n_{\mathrm{sym}}} \lambda_i^{(\mathrm{sym})} \, b_i^{(\mathrm{sym})} \, [b_i^{(\mathrm{sym})}]^T + \sum_{i=1}^{n_{\mathrm{skew}}} \lambda_i^{(\mathrm{skew})} \, b_i^{(\mathrm{skew})} \, [b_i^{(\mathrm{skew})}]^T,$$

where $\Pi_{nn} b_i^{(\mathrm{sym})} = b_i^{(\mathrm{sym})}$ and $\Pi_{nn} b_i^{(\mathrm{skew})} = -b_i^{(\mathrm{skew})}$. We may assume

$$|\lambda_1^{(\mathrm{sym})}| \geq \cdots \geq |\lambda_{n_{\mathrm{sym}}}^{(\mathrm{sym})}|$$

and

$$|\lambda_1^{(\mathrm{skew})}| \geq \cdots \geq |\lambda_{n_{\mathrm{skew}}}^{(\mathrm{skew})}|.$$

To get an unnormalized KPSVD of A, we follow (2.23) and (2.24) and reshape the eigenvectors of $\tilde{A}$ into n-by-n matrices. The sym-eigenvectors give us symmetric matrices $B_1^{(\mathrm{sym})}, \ldots, B_{n_{\mathrm{sym}}}^{(\mathrm{sym})}$ while the skew-eigenvectors give us skew-symmetric matrices $B_1^{(\mathrm{skew})}, \ldots, B_{n_{\mathrm{skew}}}^{(\mathrm{skew})}$. Overall we obtain

$$A = \sum_{i=1}^{n_{\mathrm{sym}}} \lambda_i^{(\mathrm{sym})} \, (B_i^{(\mathrm{sym})} \otimes B_i^{(\mathrm{sym})}) + \sum_{i=1}^{n_{\mathrm{skew}}} \lambda_i^{(\mathrm{skew})} \, (B_i^{(\mathrm{skew})} \otimes B_i^{(\mathrm{skew})})$$

which can be regarded as an unnormalized KPSVD of A.

2.5. A structured Cholesky-based representation. Now assume that A is PS-symmetric *and* positive semidefinite with rank r. Analogously to how we proceeded in the centrosymmetric case, we develop a structured representation of A that is based on pivoted Cholesky factorizations of the matrices $A^{(\mathrm{sym})}$ and $A^{(\mathrm{skew})}$ in (2.18) and (2.19). We compute the pivoted Cholesky factorizations

$$P^{(\mathrm{sym})} A^{(\mathrm{sym})} P^{(\mathrm{sym})T} = L^{(\mathrm{sym})} L^{(\mathrm{sym})T}, \tag{2.26}$$

$$P^{(\mathrm{skew})} A^{(\mathrm{skew})} P^{(\mathrm{skew})T} = L^{(\mathrm{skew})} L^{(\mathrm{skew})T}, \tag{2.27}$$

where

$$L^{(\mathrm{sym})} \in \mathbb{R}^{n_{\mathrm{sym}} \times r_{\mathrm{sym}}}, \qquad r_{\mathrm{sym}} = \mathrm{rank}(A^{(\mathrm{sym})}), \tag{2.28}$$

$$L^{(\mathrm{skew})} \in \mathbb{R}^{n_{\mathrm{skew}} \times r_{\mathrm{skew}}}, \qquad r_{\mathrm{skew}} = \mathrm{rank}(A^{(\mathrm{skew})}). \tag{2.29}$$

The matrices $\{L^{(\mathrm{sym})}, P^{(\mathrm{sym})}, L^{(\mathrm{skew})}, P^{(\mathrm{skew})}\}$ collectively define a structured representation of A, for if

$$\begin{aligned} Y^{(\mathrm{sym})} &= Q^{(\mathrm{sym})} P^{(\mathrm{sym})T} L^{(\mathrm{sym})} = [\, y_1^{(\mathrm{sym})} \mid \cdots \mid y_{r_{\mathrm{sym}}}^{(\mathrm{sym})} \,], \\ Y^{(\mathrm{skew})} &= Q^{(\mathrm{skew})} P^{(\mathrm{skew})T} L^{(\mathrm{skew})} = [\, y_1^{(\mathrm{skew})} \mid \cdots \mid y_{r_{\mathrm{skew}}}^{(\mathrm{skew})} \,], \end{aligned}$$

then it follows from $A = Q^{(\mathrm{sym})} A^{(\mathrm{sym})} {Q^{(\mathrm{sym})}}^T + Q^{(\mathrm{skew})} A^{(\mathrm{skew})} {Q^{(\mathrm{skew})}}^T$ that

$$A = \sum_{i=1}^{r_{\mathrm{sym}}} y_i^{(\mathrm{sym})} \, [y_i^{(\mathrm{sym})}]^T + \sum_{i=1}^{r_{\mathrm{skew}}} y_i^{(\mathrm{skew})} \, [y_i^{(\mathrm{skew})}]^T. \tag{2.30}$$

Each of the rank-1 matrices in this expansion is PS-symmetric because

$$\begin{aligned} \Pi_{nn} Y^{(\mathrm{sym})} &= (\Pi_{nn} Q^{(\mathrm{sym})})(P^{(\mathrm{sym})T} L^{(\mathrm{sym})}) \;\; = \;\; Q^{(\mathrm{sym})}(P^{(\mathrm{sym})T} L^{(\mathrm{sym})}) \;\; = Y^{(\mathrm{sym})}, \\ \Pi_{nn} Y^{(\mathrm{skew})} &= (\Pi_{nn} Q^{(\mathrm{skew})})(P^{(\mathrm{skew})T} L^{(\mathrm{skew})}) = -Q^{(\mathrm{sym})}(P^{(\mathrm{skew})T} L^{(\mathrm{skew})}) = -Y^{(\mathrm{skew})}. \end{aligned}$$

Thus, by combining the block diagonalization with pivoted Cholesky factorizations we

Representing a Positive Semidefinite PS-Symmetric A
1. Form $A^{(\mathrm{sym})}$ using (2.18).
2. Compute the pivoted Cholesky factorization of $A^{(\mathrm{sym})}$. See (2.26) and (2.28).
3. Form $A^{(\mathrm{skew})}$ using (2.19)
4. Compute the pivoted Cholesky factorization of $A^{(\mathrm{skew})}$. See (2.27) and (2.29).

FIG. 2. *Computing the representation* $\{L^{(sym)}, P^{(sym)}, L^{(skew)}, P^{(skew)}\}$ *of a PS-symmetric matrix.*

	$r_{\mathrm{sym}} = n_{\mathrm{sym}}$, $r_{\mathrm{skew}} = n_{\mathrm{skew}}$		$r_{\mathrm{sym}} = n$, $r_{\mathrm{skew}} = n$	
n	T_u/T_s	$T_{set\text{-}up}/T_s$	T_u/T_s	$T_{set\text{-}up}/T_s$
39	1.69	0.44	0.65	0.75
55	2.33	0.32	0.69	0.78
67	2.48	0.28	0.67	0.69
77	2.81	0.22	0.75	0.69

FIG. 3. T_u *is the time required to compute the Cholesky factorization of* A *(the unstructured algorithm).* T_s *is the time required to set up* $A^{(sym)}$ *and* $A^{(skew)}$ *and compute their Cholesky factorizations (the structured algorithm).* $T_{\mathrm{set\text{-}up}}$ *is the time required to just set-up* $A^{(sym)}$ *and* $A^{(skew)}$*. The LAPACK procedures POTRF (unpivoted Cholesky calling level-3 BLAS) and PSTRF (pivoted Cholesky calling level-3 BLAS) were used for full-rank and low-rank cases, respectively. Results are based on running numerous random trials for each combination of* n *and* (r_{sym}, r_{skew})*. A single core of the Intel(R) Core(TM) i5-3210M CPU @ 2.50 GHz was used.*

can efficiently represent a given positive semidefinite matrix with PS-symmetry. The procedure is summarized in Figure 2. By truncating the summations in (2.30) we can use this framework to construct low-rank approximations that are also PS-symmetric. We shall have more to say about this and related implementation issues in section 4. To anticipate the discussion we share some benchmarks in Figure 3.

The results are similar to what is reported in Table 1 for the centrosymmetric problem. In the full-rank case we anticipate a fourfold speed-up because the matrices $A^{(\mathrm{sym})}$ and $A^{(\mathrm{skew})}$ have dimensions that are about half the dimension of A. However, T_u/T_s is somewhat less than 4 because the set-up time fraction $T_{set\text{-}up}/T_s$ is nontrivial. In the low-rank case, this overhead rivals the cost of the half-size factorizations because of the reliance upon traditional right-looking procedures that force us to carry out the complete block diagonalization beforehand.

3. ((1,2),(3,4))-symmetry. We now apply the results of the previous section to the structured multilinear product (1.2). To drive the discussion we consider an example that arises in quantum chemistry and related application areas. The underlying tensor is ((1,2),(3,4)))-symmetric and its $[1,2] \times [3,4]$ unfolding is near a matrix with very low rank.

3.1. Unfolding a ((1,2),(3,4))-symmetric tensor. If $\mathcal{A} \in \mathbb{R}^{n\times n\times n\times n}$ is ((1,2),(3,4))-symmetric, then its $[1,2] \times [3,4]$ unfolding has three important properties that are tabulated in Figure 4. We refer to an n^2-by-n^2 matrix A that satisfies $A = A^T$, $\Pi_{nn}A = A$, and $A = A\Pi_{nn}$ as a ((1,2),(3,4))-*symmetric matrix*. Such a matrix is also PS-symmetric because the properties $\Pi_{nn}A = A$ and $A\Pi_{nn} = A$ imply $\Pi_{nn}A\Pi_{nn} = A$. This permits us to say a little more about the block diagonalization in (2.8).

Symmetry in $\mathcal{A}$	Implication for $A = \mathcal{A}_{[1,2]\times[3,4]}$
$\mathcal{A}(i_1,i_2,i_3,i_4) = \mathcal{A}(i_3,i_4,i_1,i_2)$	$A = A^T$
$\mathcal{A}(i_1,i_2,i_3,i_4) = \mathcal{A}(i_2,i_1,i_3,i_4)$	$\Pi_{nn}A = A$
$\mathcal{A}(i_1,i_2,i_3,i_4) = \mathcal{A}(i_1,i_2,i_4,i_3)$	$A\Pi_{nn} = A$

FIG. 4. *Unfolding a* ((1,2),(3,4))*-tensor.*

Representing a ((1,2),(3,4))-Symmetric Matrix A that is Positive Semidefinite
1. Form $A^{(\mathrm{sym})}$ using (3.1).
2. Compute the pivoted Cholesky factorization of $A^{(\mathrm{sym})}$. See (2.26) and (2.27).

FIG. 5. *Computing the representation* $\{L^{(sym)}, P^{(sym)}\}$ *of a* ((1,2),(3,4))*-symmetric matrix.*

THEOREM 3.1. *If the* n^2*-by-*n^2 *matrix* A *is* ((1,2),(3,4))*-symmetric, then*

$$Q_{nn}^T A Q_{nn} = \begin{bmatrix} A^{(sym)} & 0 \\ 0 & 0 \end{bmatrix},$$

where Q_{nn} *is defined by* (2.6)*. In other words, the diagonal block* $A^{(skew)}$ *in Theorem* 2.1 *is zero. Moreover,*

$$A^{(sym)} = \Delta^{(sym)}(u,u) \cdot A(u,u) \cdot \Delta^{(sym)}(u,u), \tag{3.1}$$

where $\Delta^{(sym)}$ *is defined by* (2.15) *and* $u = \mathtt{sym}_n$ *is given by* (2.12)*.*

Proof. Using (2.11) and the properties $A\Pi_{nn} = A$ and $\Pi_{nn}Q_{nn}^{(\mathrm{skew})} = -Q_{nn}^{(\mathrm{skew})}$, we have

$$\begin{aligned} A^{(\mathrm{skew})} &= Q_{nn}^{(\mathrm{skew})T} A Q_{nn}^{(\mathrm{skew})} = Q_{nn}^{(\mathrm{skew})T}(A\Pi_{nn})Q_{nn}^{(\mathrm{skew})} \\ &= Q_{nn}^{(\mathrm{skew})T} A(\Pi_{nn}Q_{nn}^{(\mathrm{skew})}) = -Q_{nn}^{(\mathrm{skew})T} A Q_{nn}^{(\mathrm{skew})} = -A^{(\mathrm{skew})}. \end{aligned}$$

Thus, $A^{(\mathrm{skew})} = 0$. Equation (3.1) follows by noting that $A(u,p(u)) = A(u,u)$ in (2.18). ∎

With this added bit of structure we can construct a representation that is more abbreviated than what is laid out in Figure 2 for matrices that are merely PS-symmetric. Observe in Figure 5 that only a single half-sized factorization is required. The impact of the set-up overhead in the first step is discussed in section 4.

3.2. An example. The four-index *electron repulsion integral* (ERI) tensor, also known as the two-electron integral (TEI) tensor, $\mathcal{A} \in \mathbb{R}^{n\times n\times n\times n}$ is defined by

$$\mathcal{A}(i_1,i_2,i_3,i_4) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{\phi_{i_1}(\mathbf{r}_1)\phi_{i_2}(\mathbf{r}_1)\phi_{i_3}(\mathbf{r}_2)\phi_{i_4}(\mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|} d\mathbf{r}_1 d\mathbf{r}_2, \tag{3.2}$$

where a set of basis functions $\{\phi_k\}_{1\le k\le n}$ is given such that $\phi_k \in H^1(\mathbb{R}^3)$. In general, the ϕ_k are complex basis functions but in this paper we assume real basis functions.

The simplest real basis functions ϕ_k are Gaussians parametrized by the exponents $\alpha_k \in \mathbb{R}$ and centers $\mathbf{r}_k \in \mathbb{R}^3$ for $k = 1, \ldots, n$, e.g.,

$$\phi_k(\mathbf{r}) = g_k(\mathbf{r} - \mathbf{r}_k) = (2\alpha_k/\pi)^{3/4} e^{-\alpha_k \|\mathbf{r}-\mathbf{r}_k\|^2}.$$

Typically, more sophisticated basis functions are composed from linear combinations of these simple Gaussians [5].

The ERI tensor is essential to electronic structure theory and ab initio quantum chemistry. Efficient numerical algorithms for computing and representing this tensor have been a major preoccupation for researchers interested in ab initio quantum chemistry [2, 7, 12, 9].

Notice that for the ERIs,

$$\mathcal{A}(i_1, i_2, i_3, i_4) = \mathcal{A}(i_3, i_4, i_1, i_2)$$

because the order of integration does not matter. Also, due to the commutativity of scalar multiplication,

$$\mathcal{A}(i_1, i_2, i_3, i_4) = \mathcal{A}(i_2, i_1, i_3, i_4) = \mathcal{A}(i_1, i_2, i_4, i_3) = \mathcal{A}(i_2, i_1, i_4, i_3).$$

Thus, if $\mathcal{A}$ is the tensor defined by (3.2), then it is ((1,2),(3,4))-symmetric. Moreover, $A = \mathcal{A}_{[1,2]\times[3,4]}$ is positive definite since it is a Gram matrix for the product-basis set $\{\phi_i\phi_j\}$ in the Coulomb metric $\langle \cdot, \frac{1}{\|\mathbf{r}_1-\mathbf{r}_2\|}\cdot\rangle$; see [8] for details.

3.3. A structured multilinear product. A structured version of (3.2) arises in the Hartree–Fock method, an important technique for those concerned with the *ab initio* calculation of electronic structure. Szabo and Ostlund [15] is an excellent general reference in this regard. For an accurate treatment of electronic correlation effects, it is convenient to transform the ERI tensor from the atomic orbital basis $\{\phi_k(\mathbf{r})\}$ to the molecular orbital basis $\{\psi_k(\mathbf{r})\}$. The change of basis is defined by

$$\psi_p = \sum_{q=1}^{n} X(p, q)\phi_q, \qquad p = 1, 2, \ldots, n, \tag{3.3}$$

where $X \in \mathbb{R}^{n\times n}$ is given. The goal is to transform the atomic orbital basis ERI tensor $\mathcal{A}$ into the following molecular orbital basis ERI tensor $\mathcal{B} \in \mathbb{R}^{n\times n\times n\times n}$ defined by

$$\mathcal{B}(i_1, i_2, i_3, i_4) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{\psi_{i_1}(\mathbf{r}_1)\psi_{i_2}(\mathbf{r}_1)\psi_{i_3}(\mathbf{r}_2)\psi_{i_4}(\mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|} d\mathbf{r}_1 d\mathbf{r}_2. \tag{3.4}$$

By substituting (3.3) into (3.2) it is easy to show that this tensor is given by

$$\mathcal{B}(i_1, i_2, i_3, i_4) = \sum_{j_1,j_2,j_3,j_4=1}^{n} \mathcal{A}(j_1, j_2, j_3, j_4)X(i_1, j_1)X(i_2, j_2)X(i_3, j_3)X(i_4, j_4). \tag{3.5}$$

To analyze and exploit the structure of this computation, we start with the fact that it is a special case of the general multilinear product

$$\mathcal{B}(j_1, j_2, j_3, j_4) = \sum_{i_1,i_2,i_3,i_4=1}^{n} \mathcal{A}(i_1, i_2, i_3, i_4)X_1(i_1, j_1)X_2(i_2, j_2)X_3(i_3, j_3)X_3(i_4, j_4). \tag{3.6}$$

It can be shown that

$$\mathcal{B}_{[1,2]\times[3,4]} = (X_2 \otimes X_1)\, \mathcal{A}_{[1,2]\times[3,4]}\, (X_4 \otimes X_3)^T; \tag{3.7}$$

see [6, pp. 728–729]. Thus, if

$$\mathcal{B}(i_1,i_2,i_3,i_4) = \sum_{j_1,j_2,j_3,j_4=1}^{n} \mathcal{A}(j_1,j_2,j_3,j_4) X(i_1,j_1) X(i_2,j_2) X(i_3,j_3) X(i_4,j_4), \tag{3.8}$$

then it follows that

$$\mathcal{B}_{[1,2]\times[3,4]} = (X \otimes X)\, \mathcal{A}_{[1,2]\times[3,4]}\, (X \otimes X)^T.$$

It is easy to verify that if the tensor $\mathcal{A}$ is ((1,2),(3,4))-symmetric then the tensor $\mathcal{B}$ is also ((1,2),(3,4))-symmetric. Indeed,

$$\begin{aligned} \Pi_{nn}\mathcal{B}_{[1,2]\times[3,4]} &= \Pi_{nn}(X\otimes X)\, \mathcal{A}_{[1,2]\times[3,4]}\, (X\otimes X)^T \\ &= (X\otimes X)(\Pi_{nn}\, \mathcal{A}_{[1,2]\times[3,4]})\, (X\otimes X)^T \\ &= (X\otimes X)\, \mathcal{A}_{[1,2]\times[3,4]}\, (X\otimes X)^T = \mathcal{B}_{[1,2]\times[3,4]}, \end{aligned}$$

where we used the fact that $\Pi_{nn}(M_1 \otimes M_2) = (M_2 \otimes M_1)\Pi_{nn}$ for all $M_1, M_2 \in \mathbb{R}^{n\times n}$; see [6, p. 27]. Likewise, $\mathcal{B}_{[1,2]\times[3,4]}\Pi_{nn} = \mathcal{B}_{[1,2]\times[3,4]}$. Since $\mathcal{B}_{[1,2]\times[3,4]}$ is obviously symmetric, we see that this matrix (and hence the tensor $\mathcal{B}$) is ((1,2),(3,4))-symmetric.

4. Discussion. To check out the ideas presented in the previous sections, we implemented the method displayed in Figure 5 and tested it on the low-rank ((1,2),(3,4))-symmetric matrices that arise from ERI tensor unfoldings.

4.1. Low rank. It is well known in the ERI setting that $\mathcal{A}_{[1,2]\times[3,4]}$ is very close to a matrix whose rank in $O(n)$. Indeed, Røeggen and Wisløff-Nilssen [12] show that $\text{rank}_{10^{-p}}(A) \approx pn$, where $\text{rank}_\delta(A)$ is the number of A's singular values that are greater than δ. Affirmations of this heuristic can be found in O'Neal and Simons [9] and Khoromskaia, Khoromskij, and Schneider [8]. For insight we graphically display the eigenvalue decay for two simple molecules in Figure 6. See [12] for more details on the low-rank structure.

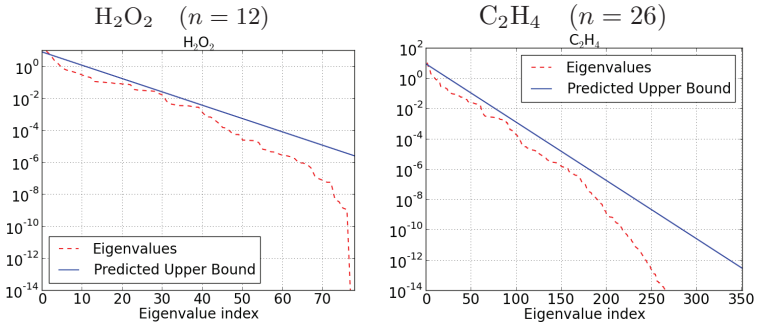

FIG. 6. *Eigenvalue decay of ERI matrices generated by the Psi4 quantum chemistry package* [16].

4.2. A lazy evaluation strategy. Beebe and Linderberg [2] were the first to show that by making use of the low rank and positive definiteness of the two-electron integral matrix it is possible to reduce the number of integral evaluations necessary to factorize the matrix, as well as reduce the complexity of a major bottleneck of computational quantum chemistry called the TEI four-index transformation. The key idea is to implement the pivoted Cholesky factorization algorithm with lazy evaluation–off-diagonal entries (integrals) that are only computed when necessary. To illustrate, after (say) k steps of the process on an N-by-N matrix A, we have the following partial factorization

$$P_k A P_k^T = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{N-k} \end{bmatrix} \begin{bmatrix} I_k & 0 \\ 0 & \tilde{A} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{N-k} \end{bmatrix}^T. \tag{4.1}$$

Ordinarily, the matrix $\tilde{A}$ is fully available, its diagonal is scanned for the largest entry, and then a PAP^T-type of permutation update is performed that brings this largest diagonal entry to the $(k+1, k+1)$ position. The step is completed by carrying a rank-1 update of the permuted $\tilde{A}$ and this renders the next column of L. The lazy evaluation version of this recognizes that we do not need the off-diagonal values in $\tilde{A}$ to determine the pivot. *Only the diagonal of $\tilde{A}$ is necessary to carry out the pivot strategy.* The recipe for the next column of L involves (a) previously computed columns of L and (b) entries from that column of A which is associated with the pivot. It is then an easy matter to update the diagonal of the current $\tilde{A}$ to get the diagonal of the "next" $\tilde{A}$. The importance of this lazy-evaluation strategy is that $O(Nk)$ integral evaluations (i.e., a_{ij} evaluations) are necessary to get through the kth step. If the largest diagonal entry in $\tilde{A}$ is less than a small tolerance δ, then because $\tilde{A}$ is positive definite, $\| \tilde{A} \| = O(\delta)$ and we have the "right" to regard A as a rank-k matrix. The overall technique can be seen as a combination of Gaxpy-Cholesky, which only needs $A(k{:}n, k)$ in step k and outer product Cholesky which is traditionally used in situations that involve diagonal pivoting. Røeggen and Wisløff-Nilssen [12] explore the numerical rank of the TEI matrix, and investigate the relationship of various thresholds and electronic properties; see also [7, 8].

While on the subject of lazy evaluation, it is important to stress that the matrix entries in $A^{(\mathrm{sym})}$ are essentially entries from A; see (3.2). Thus, when we apply our implementation of pivoted Cholesky to $A^{(\mathrm{sym})}$ with lazy evaluation, there are no extra a_{ij} computations. In other words, our method requires half the work, half the storage, and half the electronic repulsion integrals as traditional Cholesky-based methods. Table 2 confirms these observations

4.3. Computational savings due to multiple symmetries versus low rank. As we observed in the previous subsection, when A is an $n^2 \times n^2$ matrix with ((1,2),(3,4))-symmetry and rank $r \ll n^2$, then we only reduce the factorization time by roughly a factor of 2. This phenomenon can be explained as follows. If unstructured pivoted Cholesky on a low-rank $n^2 \times n^2$ matrix with ((1,2),(3,4))-symmetry takes $c(n^2)r^2$ flops, and structured pivoted Cholesky on the same matrix (which corresponds to pivoted Cholesky on a low-rank approximately $(n^2/2) \times (n^2/2)$ matrix) takes approximately $c(n^2/2)r^2 = (c(n^2)r^2)/2$ flops, then the structured algorithm can only reduce the amount of work by a factor of 2. In summary, when we use a rank-revealing Cholesky algorithm on a low-rank matrix with ((1,2),(3,4))-symmetry, we expect 2 times less work than for a low-rank matrix without this symmetry.

However, consider when the matrix A is an $n^2 \times n^2$ matrix with ((1,2),(3,4))-symmetry and full possible rank (rank $r = n(n+1)/2 \approx n^2/2$, since there are only

TABLE 2

T_u is the time in seconds to compute the lazy-evaluation pivoted Cholesky factorization of A (the unstructured algorithm). T_s the time in seconds to compute the lazy-evaluation pivoted Cholesky factorization of $A(u,u)$ (the structured algorithm). S_u is the number of bytes allocated to factorize A (unstructured). S_s is the number of bytes allocated to factorize $A(u,u)$ (structured). E_u is the number of ERI evaluations to factorize A (unstructured). E_s is number of ERI evaluations to factorize $A(u,u)$ (structured). Results are based on running Psi4 lazy-evaluation pivoted Cholesky on the ERI matrix of four different molecules on a single core of a laptop Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz. Subscript u refers to "unstructured pivoted Cholesky algorithm which doesn't utilize multiple symmetries" while subscript s refers to "structured pivoted Cholesky algorithm which does utilize multiple symmetries".

	$n = 44$ $r = 345$	$n = 72$ $r = 560$	$n = 88$ $r = 720$	$n = 116$ $r = 918$
T_u/T_s	1.84	1.90	1.89	1.93
S_u/S_s	1.95	1.97	1.97	1.98
E_u/E_s	1.95	1.97	1.97	1.98

$n(n+1)/2$ unique columns). If unstructured Cholesky without pivoting takes $c(n^2)^3 = cn^6$ flops, and structured Cholesky without pivoting (which corresponds to Cholesky without pivoting on an approximately $(n^2/2) \times (n^2/2)$ matrix) takes approximately $c(n^2/2)^3 = (cn^6)/8$ flops, then the structured algorithm can reduce the work required for the Cholesky factorization by a factor of 8. In summary, when we use a nonrank-revealing Cholesky algorithm on a matrix with ((1,2),(3,4))-symmetry, we expect 8 times less work compared to a matrix without this symmetry.

5. Conclusion. We have used a simple example of multiple symmetries to explore a computational framework that involves block diagonalization and the pivoted Cholesky factorization. Items on our research agenda include the extension of these ideas to more intricate forms of multiple symmetry that arise in higher-order tensor problems and to apply this approach to improve the performance of the Hartree–Fock method in quantum chemistry. Intelligent data structures and blocking will certainly be part of the picture. Ragnarsson and Van Loan develop a block tensor computation framework in [11]. If multiple symmetries are present then, as in the matrix case, tensions arise between compact storage schemes and "layout friendly" matrix multiplication formulations; see Epifanovsky et al. [4] and Solomonik et al. [14]. In [13] Schatz et al. discuss a blocked data structure for symmetric tensors, partial symmetry, and the prospect of building a general purpose library for multilinear algebra computation. They also discuss a blocking strategy for a symmetric multilinear product.

Acknowledgments. J.P.V. thanks Todd J. Martinez for useful conversations about quantum chemistry. The authors wish to thank the referees for their useful suggestions.

REFERENCES

[1] A. L. ANDREW, *Classroom note: Centrosymmetric matrices*, SIAM Rev., 40 (1998), pp. 697–698.

[2] N. H. F. BEEBE AND J. LINDERBERG, *Simplifications in the generation and transformation of two-electron integrals in molecular calculations*, Internat. J. Quantum Chem., 12 (1977), pp. 683–705.

[3] L. DATTA AND S. D. MORGERA, *On the reducibility of centrosymmetric matrices applications in engineering problems*, Circuits Systems Signal Process., 8 (1989), pp. 71–96.

[4] E. Epifanovsky, M. Wormit, T. Kuś, A. Landau, D. Zuev, K. Khistyaev, P. Manohar, I. Kaliman, A. Dreuw, and A. I Krylov, *New implementation of high-level correlated methods using a general block tensor library for high-performance electronic structure calculations*, J. Comput. Chem., 34 (2013), pp. 2293–2309.

[5] P. M. W. Gill, M. Head-Gordon, and J. A. Pople, *An efficient algorithm for the generation of two-electron repulsion integrals over Gaussian basis functions*, Internat. J. Quantum Chem., 36 (1989), pp. 269–280.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, MD, 2013.

[7] H. Harbrecht, M. Peters, and R. Schneider, *On the low-rank approximation by the pivoted Cholesky decomposition*, Appl. Numer. Math., 62 (2012), pp. 428–440.

[8] V. Khoromskaia, B. N. Khoromskij, and R. Schneider, *Tensor-structured factorized calculation of two-electron integrals in a general basis*, SIAM J. Sci. Comput., 35 (2013), pp. A987–A1010.

[9] D. W O'Neal and J. Simons, *Application of Cholesky-like matrix decomposition methods to the evaluation of atomic orbital integrals and integral derivatives*, Internat. J. Quantum Chem., 36 (1989), pp. 673–688.

[10] I. S. Pressman, *Matrices with multiple symmetry properties: Applications of centrohermitian and perhermitian matrices*, Linear Algebra Appl., 284 (1998), pp. 239–258.

[11] S. Ragnarsson and C. F. Van Loan, *Block tensor unfoldings*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 149–169.

[12] I. Røeggen and E. Wisløff-Nilssen, *On the Beebe-Linderberg two-electron integral approximation*, Chem. Phys. Lett., 132 (1986), pp. 154–160.

[13] M. D. Schatz, T. M. Low, R. A. Van De Geijn, and T. G. Kolda, *Exploiting symmetry in tensors for high performance: Multiplication with symmetric tensors*, SIAM J. Sci. Comput., 36 (2014), pp. C453-C479.

[14] E. Solomonik, D. Matthews, J. Hammond, and J. Demmel, *Cyclops tensor framework: Reducing communication and eliminating load imbalance in massively parallel contractions*, in IEEE 27th International Symposium on Parallel & Distributed Processing (IPDPS), 2013, IEEE, Piscataway, NJ, 2013, pp. 813–824.

[15] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, McGraw-Hill, New York, 1989.

[16] J. M. Turney et al., *Psi*4: *An open-source ab initio electronic structure program*, Wiley Interdiscip. Rev. Comput. Mol. Sci., 2 (2012), pp. 556–565.

[17] C. F. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, Front. Appl. Math. 10, SIAM, Philadelphia, 1992.

[18] C. F. Van Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.

[19] C. F. Van Loan and N. Pitsianis, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real-Time Applications, Kluwer Academic, Dordrecht, 1993, pp. 293–314.