Abstract. This paper presents the *derivation* of a new algorithm for the stable computation of sample partial correlation coefficients.

We start with Bareiss' algorithm for the solution of linear systems of equations with (nonsymmetric) Toeplitz coefficient matrix and show how to generalize it to matrices that are not Toeplitz. The so generalized Bareiss algorithm computes the LU and UL factorizations of those matrices whose contiguous principal submatrices are all non-singular. For symmetric positive-definite matrices A, which naturally satisfy this condition, the normalized version of Bareiss' algorithm is just the Hyperbolic Cholesky algorithm, which computes the upper and lower triangular Cholesky factors U and L of A by means of 2×2 hyperbolic rotations.

Guided by the data flow graph of the Hyperbolic Cholesky algorithm, we show that there exists one sequence of 2×2 orthogonal rotations that effects the transformation from U to L such that the sines of these rotations equal the hyperbolic tangents of the hyperbolic rotations. From the connection to the Hyperbolic Cholesky algorithm it also follows that, if A is a sample covariance matrix, the sines are sample partial correlation coefficients.

Consequently, given a data matrix B it is recommended to determine sample partial correlations directly from the columns of B instead of forming $A = B^T B$ and inviting potential loss of numerical accuracy: compute the QR decomposition $B = QU$; apply plane rotations to transform U to L; the sines of the rotation angles are partial correlations.

From Bareiss' Algorithm to the Stable Computation of Partial Correlations

Jean-Marc Delosme[1], Ilse C.F. Ipsen[2]

Research Report YALEU/DCS/RR-624
May 1988

[1] Department of Electrical Engineering, Yale University
[2] Department of Computer Science, Yale University

The work presented in this paper was supported by the Office of Naval Research under contracts N000014-86-K-0310 and N00014-85-K-0461, and by the Army Research Office under contract DAAL03-86-K-0158. Approved for public release: distribution is unlimited.

1. Introduction

In [6, 7] a new algorithm was introduced for the stable computation of sample partial correlation coefficients: the partial correlations are computed directly from the data matrix so as to avoid the loss of numerical accuracy typically associated with the formation of the sample covariance matrix (numerical examples are given in [6, 7]). This paper recounts the context that led to the discovery of this result and presents derivations of the algorithm from multiple vantage points.

In Section 2 we start with Bareiss' algorithm for the solution of linear systems of equations with (non-symmetric) Toeplitz coefficient matrix [3] and show how to generalize it to matrices that are not Toeplitz. The so generalized Bareiss algorithm computes the LU and UL factorizations of those matrices whose contiguous principal submatrices are all non-singular. For symmetric positive-definite matrices A, which naturally satisfy this condition, Section 3 shows that the normalized version of Bareiss' algorithm is just the Hyperbolic Cholesky algorithm [5], which computes the upper and lower triangular Cholesky factors U and L of A by means of 2×2 hyperbolic rotations.

Guided by the data flow graph of the Hyperbolic Cholesky algorithm, we show that there exists one sequence of 2×2 orthogonal rotations that effects the transformation from U to L such that the sines of these rotations equal the hyperbolic tangents of the hyperbolic rotations. From the connection to the Hyperbolic Cholesky algorithm it also follows that, if A is a sample covariance matrix, the sines are sample partial correlation coefficients. Section **paralg** provides both, algebraic and a geometric derivations of this result.

Since, given a data matrix B, sample partial correlations can be determined in a stable way from a Cholesky factor of $A = B^T B$ it is recommended to determine them directly from the columns of B instead of forming A and inviting potential loss of numerical accuracy: compute the QR decomposition $B = QU$; apply plane rotations to transform U to L; the sines of the rotation angles are partial correlations. More specifically, the transformation of U to L yields all partial correlations between variables B_i and B_{i+k}, given the variables inbetween: $B_{i+1} \ldots B_{i+k-1}$. Section **arbpar** concludes the paper with some remarks on how to go about computing sample partial correlations given arbitrary sets of variables.

2. The Generalized Bareiss Algorithm

In [3] E. Bareiss proposed an algorithm to solve the linear system

$$Ax = b,$$

where A is a $n \times n$ Toeplitz matrix; A does not have to be symmetric.

2.1. Description of the Algorithm

The algorithm reduces A to triangular form, modifies the right-hand side b accordingly, and determines x by backsubstitution. The reduction process operates on the $2n \times n$ array

$$\begin{pmatrix} A \\ A \end{pmatrix},$$

and Bareiss proposes two versions of this process (see Sections 2 and 3 in [3]). We will generalize the second version that treats the upper and lower halves of the array symmetrically, and merely ignore the condition that A be Toeplitz.

The reduction process proceeds by removing successive superdiagonals in the upper half of the array and successive subdiagonals in the lower half of the array. At completion, the array is of the form

$$\begin{pmatrix} L \\ U \end{pmatrix},$$

where L is a lower triangular matrix and U is an upper triangular matrix.

Before providing a formal description of the generalized Bareiss algorithm, we will illustrate the algorithm by considering the case $n = 4$. The initial array is of the form

$$\begin{pmatrix} A \\ A \end{pmatrix} = \left(\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ l & l & l & l \\ \hline u & u & u & u \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{array}\right).$$

The stars indicate the elements to be modified in step 1, the middle two rows remain unchanged. The reduction process proceeds as follows.

1. Linear combinations of row i in the upper half of the array with row $i+1$ in the lower half of the array eliminate element $(i, i+1)$ in the upper half and element $(i+1, i)$ in the lower half, $1 \leq i \leq 3$:

$$\left(\begin{array}{cccc} d & s & * & * \\ * & d & s & * \\ * & * & d & s \\ l & l & l & l \\ \hline u & u & u & u \\ s & d & * & * \\ * & s & d & * \\ * & * & s & d \end{array}\right) \xrightarrow{\text{step 1}} \left(\begin{array}{cccc} * & 0 & * & * \\ * & * & 0 & * \\ * & * & * & 0 \\ l & l & l & l \\ \hline u & u & u & u \\ 0 & * & * & * \\ * & 0 & * & * \\ * & * & 0 & * \end{array}\right),$$

   Super- or subdiagonal elements doomed for elimination are represented by s, and diagonal elements participating in the elimination by d. Appropriate multiples for the linear combinations are made up from the ratios s/d.

2. Linear combinations of row i in the upper half of the array with row $i+2$ in the lower half of the array eliminate element $(i, i+2)$ in the upper half and element $(i+2, i)$ in the lower half, $i = 1, 2$:

$$\left(\begin{array}{cccc} d & 0 & s & * \\ * & d & 0 & s \\ l & l & l & 0 \\ l & l & l & l \\ \hline u & u & u & u \\ 0 & u & u & u \\ s & 0 & d & * \\ * & s & 0 & d \end{array}\right) \xrightarrow{\text{step 2}} \left(\begin{array}{cccc} * & 0 & 0 & * \\ * & * & 0 & 0 \\ l & l & l & 0 \\ l & l & l & l \\ \hline u & u & u & u \\ 0 & u & u & u \\ 0 & 0 & * & * \\ * & 0 & 0 & * \end{array}\right)$$

   Note that the two rows responsible for an elimination contain zeros in corresponding positions, hence no previously introduced zeros are destroyed.

3. Linear combinations of row 1 in the upper half of the array with row 4 in the lower half of the

array eliminate element $(1,4)$ in the upper half and element $(4,1)$ in the lower half:

$$\left(\begin{array}{cccc} d & 0 & 0 & s \\ l & l & 0 & 0 \\ l & l & l & 0 \\ l & l & l & l \\ \hline u & u & u & u \\ 0 & u & u & u \\ 0 & 0 & u & u \\ s & 0 & 0 & d \end{array}\right) \xrightarrow{\text{step 3}} \left(\begin{array}{cccc} * & 0 & 0 & 0 \\ l & l & 0 & 0 \\ l & l & l & 0 \\ l & l & l & l \\ \hline u & u & u & u \\ 0 & u & u & u \\ 0 & 0 & u & u \\ 0 & 0 & 0 & * \end{array}\right).$$

The matrices L and U obtained through this process will be shown in Section usefac to be the right factors in the UL and LU decompositions of A. The total number of operations to find both decompositions comes to $2\sum_{i=1}^{n-1}(n-i) = n(n-1)$ divisions and $2\sum_{i=1}^{n-1}(n-i)^2 = \frac{2}{3}n(n-\frac{1}{2})(n-1)$ multiplications; this is twice the number of operations for Gaussian elimination without pivoting, which determines only one of the decompositions.

In the formal description of the generalized Bareiss algorithm let, at the start of step k, $a_i^{(k-1)}$ denote row i in the upper half of the array, and $a_i^{(-k+1)}$ row i in the lower half of the array. Thus $a_i^{(+0)} = a_i^{(-0)} = a_i$ equals the ith row of A. The jth element in row $a_i^{(k-1)}$ is denoted by $a_{i,j}^{(k-1)}$, and the jth element in row $a_i^{(-k+1)}$ by $a_{i,j}^{(-k+1)}$. The reduction process may now be expressed as:

Generalized Bareiss Algorithm

$$1 \le i \le n, \quad \begin{pmatrix} a_i^{(+0)} \\ a_i^{(-0)} \end{pmatrix} = \begin{pmatrix} a_i \\ a_i \end{pmatrix}$$

for $k = 1$ to $n-1$,

$$1 \le i \le n-k, \quad \begin{pmatrix} a_i^{(k)} \\ a_{i+k}^{(-k)} \end{pmatrix} = \begin{pmatrix} 1 & -a_{i,i+k}^{(k-1)} / a_{i+k,i+k}^{(-k+1)} \\ -a_{i+k,i}^{(-k+1)} / a_{i,i}^{(k-1)} & 1 \end{pmatrix} \begin{pmatrix} a_i^{(k-1)} \\ a_{i+k}^{(-k+1)} \end{pmatrix}.$$

Note, that we attempt to distinguish parallelism in the algorithms by employing two different notations for repetitive operations: *for-loops*, such as the k-loop above, increase in unit increments and iterations are performed one after the other; *ranges*, like $1 \le i \le n-k$, denote independent iterations that can all be performed in parallel for a given k.

2.2. UL and LU Matrix Factorizations

In order to show that the algorithm computes the right factors in the UL and LU decompositions of the matrix A it is convenient to regroup, for each step k, all the operations performed on pairs of rows into a single matrix operation. Let $R^{(k)}$ denote the rectangle at the start of step k, $k \ge 1$, that consists of the 'active' rows $1 \ldots n-k$ in the upper half of the array. Similarly, $R^{(-k)}$ denotes the rectangle at the start of step k that consists of the active rows $k+1 \ldots n$ in the lower half of the array.

There are two main diagonals in a rectangular matrix: one starting from the top left corner, and the second ending at the bottom right corner. As k increases, the leftmost (top) main diagonal in $R^{(k)}$ and the rightmost (bottom) main diagonal in $R^{(-k)}$ shrink in place, they are denoted by $D^{(k)}$ and $D^{(-k)}$, respectively (cf. the d elements in the example). These diagonals are also portions of the main diagonals in the $2n \times n$ array. The diagonal matrices associated with the two remaining main diagonals in $R^{(k)}$ and $R^{(-k)}$ are denoted $S^{(k)}$ and $S^{(-k)}$, respectively (cf. the s elements in the example). Within the array these diagonals are portions of the kth superdiagonal and of the kth subdiagonal, respectively. The multipliers in the reduction process are the respective ratios of the diagonal elements of $S^{(k)}$ and $S^{(-k)}$ (i.e. the elements to be eliminated in $R^{(k)}$ and $R^{(-k)}$) and the diagonal elements of $D^{(-k)}$ and $D^{(k)}$ (the pivots). Thus, at the start of step k, $k \geq 1$,

$$D^{(k)} = \begin{pmatrix} a_{1,1}^{(k-1)} & & & \\ & a_{2,2}^{(k-1)} & & \\ & & \ddots & \\ & & & a_{n-k,n-k}^{(k-1)} \end{pmatrix}, \quad S^{(k)} = \begin{pmatrix} a_{1,k+1}^{(k-1)} & & & \\ & a_{2,k+2}^{(k-1)} & & \\ & & \ddots & \\ & & & a_{n-k,n}^{(k-1)} \end{pmatrix},$$

and

$$S^{(-k)} = \begin{pmatrix} a_{k+1,1}^{(-k+1)} & & & \\ & a_{k+2,2}^{(-k+1)} & & \\ & & \ddots & \\ & & & a_{n,n-k}^{(-k+1)} \end{pmatrix}, \quad D^{(-k)} = \begin{pmatrix} a_{k+1,k+1}^{(-k+1)} & & & \\ & a_{k+2,k+2}^{(-k+1)} & & \\ & & \ddots & \\ & & & a_{n,n}^{(-k+1)} \end{pmatrix}.$$

Then, assuming $D^{(k)}$ and $D^{(-k)}$ to be non-singular, step k consists of replacing the rectangles

$$\begin{pmatrix} R^{(k)} \\ R^{(-k)} \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} I_{n-k} & -S^{(k)}(D^{(-k)})^{-1} \\ -S^{(-k)}(D^{(k)})^{-1} & I_{n-k} \end{pmatrix} \begin{pmatrix} R^{(k)} \\ R^{(-k)} \end{pmatrix}.$$

We have to show that the product of the multiplier matrices can be appropriately partitioned into upper and lower triangular matrices. In order to examine the band structure of matrices we distinguish the two outermost non-zero diagonals of a matrix: Let the kth diagonal of a matrix M, consisting of elements $m_{i,i+k}$, have *index* k where k ranges from $-(n-1)$ to $n-1$. The *valuation* of a matrix M is defined as the smallest index $\nu(M)$ of a non-zero diagonal in M; the *degree* of M is the highest index $\delta(M)$ of a non-zero diagonal in M. Denote by $\mathbf{B}_{\nu,\delta}$ the class of $n \times n$ matrices with $\nu(M) \geq \nu$ and $\delta(M) \leq \delta$. For instance, $\mathbf{B_{0,n-1}}$ is the class of upper triangular matrices, $\mathbf{B_{-(n-1),0}}$ the class of lower triangular matrices, and $\mathbf{B_{1,n-1}}$ ($\mathbf{B_{-(n-1),-1}}$) the class of strictly upper (lower) triangular matrices.

From the multiplication and addition of band matrices one knows: if $M_1 \in \mathbf{B}_{\nu_1,\delta_1}$ and $M_2 \in \mathbf{B}_{\nu_2,\delta_2}$ then the product $M_1 M_2 \in \mathbf{B}_{\nu_1+\nu_2,\delta_1+\delta_2}$ and the sum $M_1 + M_2 \in \mathbf{B}_{\min\{\nu_1,\nu_2\},\max\{\delta_1,\delta_2\}}$. When we just want to emphasize the band structure of a matrix M in the class $\mathbf{B}_{\nu,\delta}$, we write $M_{\nu,\delta}$ in place of M. Thus, $M_{\nu_1,\delta_1} M_{\nu_2,\delta_2} = M_{\nu_1+\nu_2,\delta_1+\delta_2}$ and $M_{\nu_1,\delta_1} + M_{\nu_2,\delta_2} = M_{\min\{\nu_1,\nu_2\},\max\{\delta_1,\delta_2\}}$. Of course cancellations could occur and the indices ν and δ in $M_{\nu,\delta}$ may be smaller than the valuation of M and larger than its degree, respectively.

The transformation performed on the $2n \times n$ array at step k is a premultiplication by the $2n \times 2n$ multiplier matrix

$$\left(\begin{array}{cc|cc} I_{n-k} & & & -S^{(k)}(D^{(-k)})^{-1} \\ & I_k & 0 & \\ \hline & 0 & I_k & \\ -S^{(-k)}(D^{(k)})^{-1} & & & I_{n-k} \end{array}\right),$$

where $-S^{(k)}(D^{(-k)})^{-1}$ and $-S^{(-k)}(D^{(k)})^{-1}$ are diagonal of order $n-k$. Hence this transformation has the structure

$$\begin{pmatrix} M_{0,0} & M_{k,k} \\ M_{-k,-k} & M_{0,0} \end{pmatrix}.$$

It will now be shown by induction that the product of the transformations from step 1 to step k, $1 \leq k \leq n-1$, is of the form

$$\begin{pmatrix} M_{0,k-1} & M_{1,k} \\ M_{-k,-1} & M_{-(k-1),0} \end{pmatrix}.$$

The above representation is obviously true for $k=1$. Assume now that $2 \leq k \leq n-1$ and that the product of the transformations from step 1 to step $k-1$ is

$$\begin{pmatrix} M_{0,k-2} & M_{1,k-1} \\ M_{-(k-1),-1} & M_{-(k-2),0} \end{pmatrix}.$$

The product of the transformations from step 1 to step k is then

$$\begin{aligned} \begin{pmatrix} M_{0,0} & M_{k,k} \\ M_{-k,-k} & M_{0,0} \end{pmatrix} \begin{pmatrix} M_{0,k-2} & M_{1,k-1} \\ M_{-(k-1),-1} & M_{-(k-2),0} \end{pmatrix} &= \begin{pmatrix} M_{0,k-2} + M_{1,k-1} & M_{1,k-1} + M_{2,k} \\ M_{-k,-2} + M_{-(k-1),-1} & M_{-(k-1),-1} + M_{-(k-2),0} \end{pmatrix} \\ &= \begin{pmatrix} M_{0,k-1} & M_{1,k} \\ M_{-k,-1} & M_{-(k-1),0} \end{pmatrix}. \end{aligned}$$

Consequently, the product of the transformations from step 1 to step n is of the form

$$\begin{pmatrix} \tilde{U}_{+} & \tilde{U}_{\#} \\ \tilde{L}_{\#} & \tilde{L}_{+} \end{pmatrix},$$

where $\tilde{U}_{+}$ and $\tilde{L}_{+}$ are respective $n \times n$ upper and lower triangular matrices, and $\tilde{U}_{\#}$ and $\tilde{L}_{\#}$ are respective $n \times n$ strictly upper and lower triangular matrices. A slightly more refined induction proof exploiting the fact that the $M_{0,0}$ submatrices in the kth transformation are actually $n \times n$ identity matrices shows that the product of transformations from step 1 to step k has the form

$$\begin{pmatrix} I_n + M_{1,k-1} & M_{1,k} \\ M_{-k,-1} & I_n + M_{-(k-1),-1} \end{pmatrix},$$

so that $\tilde{U}_{+}$ and $\tilde{L}_{+}$ have unit diagonal.

Thus, the generalized Bareiss algorithm effects

$$\begin{pmatrix} \tilde{U}_{+} & \tilde{U}_{\#} \\ \tilde{L}_{\#} & \tilde{L}_{+} \end{pmatrix} \begin{pmatrix} A \\ A \end{pmatrix} = \begin{pmatrix} (\tilde{U}_{+} + \tilde{U}_{\#})A \\ (\tilde{L}_{+} + \tilde{L}_{\#})A \end{pmatrix} \equiv \begin{pmatrix} L \\ U \end{pmatrix},$$

where L is $n \times n$ lower triangular and U $n \times n$ upper triangular. Since $\tilde{U}_{+} + \tilde{U}_{\#}$ and $\tilde{L}_{+} + \tilde{L}_{\#}$ are respectively unit lower and upper triangular matrices, the matrices L and U constitute the right factors in the LU and UL decompositions of A:

$$\begin{aligned} A &= \tilde{U}L, & \tilde{U} &\equiv (\tilde{U}_{+} + \tilde{U}_{\#})^{-1} \\ A &= \tilde{L}U, & \tilde{L} &\equiv (\tilde{L}_{+} + \tilde{L}_{\#})^{-1} \end{aligned}.$$

2.3. Conditions for Successful Factorization

If the matrices $D^{(k)}$ and $D^{(-k)}$, $1 \le k \le n-1$, are non-singular then the generalized Bareiss algorithm runs to completion and delivers LU and UL decompositions of A (a necessary and sufficient condition can be derived assuming the ratio s/d is set to zero whenever $s = d = 0$). The matrices $D^{(k)}$ and $D^{(-k)}$ are non-singular when their diagonal elements

$$a_{11}^{(k-1)},\ a_{22}^{(k-1)},\ \ldots,\ a_{n-k,n-k}^{(k-1)}, \qquad 1 \le k \le n-1$$

and

$$a_{k+1,k+1}^{(-k+1)},\ a_{k+2,k+2}^{(-k+1)},\ \ldots,\ a_{n,n}^{(-k+1)}, \qquad 1 \le k \le n-1$$

are all non-zero. We shall now interpret this condition in terms of the original matrix A.

The contiguous principal submatrix

$$\begin{pmatrix} a_{i+1,i+1} & \cdots & a_{i+1,i+k-1} \\ \vdots & & \vdots \\ a_{i+k-1,i+1} & \cdots & a_{i+k-1,i+k-1} \end{pmatrix}$$

of order $k-1$ of A is called the *$(k-1)$-block after (row) i in A* or *the $(k-1)$-block before (row) $i+k$ in A*. The row vectors $a_i^{(k-1)}$ and $a_{i+k}^{(-k+1)}$ in the generalized Bareiss algorithm at step k can be expressed in terms of the inverse of the $(k-1)$-block after i in A:

$$a_i^{(k-1)} = a_i - \begin{pmatrix} a_{i,i+1} & \cdots & a_{i,i+k-1} \end{pmatrix} \begin{pmatrix} a_{i+1,i+1} & \cdots & a_{i+1,i+k-1} \\ \vdots & & \vdots \\ a_{i+k-1,i+1} & \cdots & a_{i+k-1,i+k-1} \end{pmatrix}^{-1} \begin{pmatrix} a_{i+1} \\ \vdots \\ a_{i+k-1} \end{pmatrix},$$

and

$$a_{i+k}^{(-k+1)} = a_{i+k} - \begin{pmatrix} a_{i+k,i+1} & \cdots & a_{i+k,i+k-1} \end{pmatrix} \begin{pmatrix} a_{i+1,i+1} & \cdots & a_{i+1,i+k-1} \\ \vdots & & \vdots \\ a_{i+k-1,i+1} & \cdots & a_{i+k-1,i+k-1} \end{pmatrix}^{-1} \begin{pmatrix} a_{i+1} \\ \vdots \\ a_{i+k-1} \end{pmatrix}.$$

Thus $a_i^{(k-1)}$ represents the ith row of the Schur complement of the $(k-1)$-block after i in A, and $a_{i+k}^{(-k+1)}$ the $(i+k)$th row. The Schur complement, as defined as by Ando [2], is a $n \times n$ matrix with rows $i \ldots i+k$ equal to 0 and columns $i \ldots i+k$ equal to 0.

The two equations above constitute a direct consequence of the quotient property of Schur complements (Theorem 3 in [2]). Indeed the property is true for $k = 1$, since it states $a_i^{(+0)} = a_i^{(-0)} = a_i$. Assuming it is true for the vectors $a_i^{(k-1)}$ and $a_{i+k}^{(-k+1)}$, the right-hand side in Bareiss' first recurrence relation

$$a_i^{(k)} = a_i^{(k-1)} - a_{i,i+k}^{(k-1)} (a_{i+k,i+k}^{(-k+1)})^{-1} a_{i+k}^{(-k+1)}$$

is row i of the Schur complement of the 1-block after $i+k-1$ in the Schur complement of the $(k-1)$-block after i in A and, from the quotient property, this is row i of the Schur complement of the k-block after i in A:

$$a_i^{(k)} = a_i - \begin{pmatrix} a_{i,i+1} & \cdots & a_{i,i+k} \end{pmatrix} \begin{pmatrix} a_{i+1,i+1} & \cdots & a_{i+1,i+k} \\ \vdots & & \vdots \\ a_{i+k,i+1} & \cdots & a_{i+k,i+k} \end{pmatrix}^{-1} \begin{pmatrix} a_{i+1} \\ \vdots \\ a_{i+k} \end{pmatrix}.$$

Similarly the right-hand side in the second recurrence relation

$$a^{(-k)}_{i+k} = a^{(-k+1)}_{i+k} - a^{(-k+1)}_{i+k,i}(a^{(k-1)}_{i,i})^{-1}a^{(k-1)}_{i}$$

must be row $i+k$ of the Schur complement of the k-block before $i+k$ in A:

$$a^{(-k)}_{i+k} = a_{i+k} - \begin{pmatrix} a_{i+k,i} & \ldots & a_{i+k,i+k-1} \end{pmatrix} \begin{pmatrix} a_{i,i} & \ldots & a_{i,i+k-1} \\ \vdots & & \vdots \\ a_{i+k-1,i} & \ldots & a_{i+k-1,i+k-1} \end{pmatrix}^{-1} \begin{pmatrix} a_i \\ \vdots \\ a_{i+k-1} \end{pmatrix}.$$

The interpretation of the intermediate quantities in the algorithm is now clear: $a^{(k)}_{ij}$ denotes the jth element of row i in the Schur complement of the principal submatrix of order k just *after* row i in A, similarly $a^{(-k)}_{ij}$ is the jth element of row i in the Schur complement of the principal submatrix of order k just *before* row i in A.

Clearly the condition that

$$a^{(k-1)}_{11},\ a^{(k-1)}_{22},\ \ldots,\ a^{(k-1)}_{n-k,n-k}, \qquad 1 \le k \le n-1$$

all be non-zero is equivalent to the condition that the products

$$\prod_{j=1}^{k} a^{(k-j)}_{j,j},\ \prod_{j=1}^{k} a^{(k-j)}_{1+j,1+j},\ \ldots,\ \prod_{j=1}^{k} a^{(k-j)}_{n-1-k+j,n-1-k+j}, \qquad 1 \le k \le n-1$$

all be non-zero. Likewise,

$$a^{(-k+1)}_{k+1,k+1},\ a^{(-k+1)}_{k+2,k+2},\ \ldots,\ a^{(-k+1)}_{n,n}, \qquad 1 \le k \le n-1$$

are non-zero if and only if

$$\prod_{j=1}^{k} a^{(-j+1)}_{1+j,1+j},\ \prod_{j=1}^{k} a^{(-j+1)}_{2+j,2+j},\ \ldots,\ \prod_{j=1}^{k} a^{(-j+1)}_{n-1-k+j,n-1-k+j}, \qquad 1 \le k \le n-1$$

are non-zero. The quotient property and Schur's determinantal formula (Section 4 in [4]) imply that the above products are equal to the respective *determinants* of the $(k-1)$-blocks after $1,\ 2,\ ,\ldots,\ n-k$, and before $k+1,\ k+2,\ \ldots,\ n$ for $1 \le k \le n-1$. If the determinants of the first (second) set of blocks are non-zero then all contiguous principal submatrices of orders $1\ \ldots\ n-1$ of the leading (trailing) principal submatrix of A of order $n-1$ are non-singular, hence all contiguous principal submatrices of A of orders $1\ \ldots\ n-1$ are non-singular. Note that this condition is significantly more stringent than if Gaussian elimination without pivoting were applied twice to compute U and L, then only all leading and trailing principal submatrices of orders $1\ \ldots\ n-1$ should be non-singular.

3. The Hyperbolic Cholesky Algorithm

From now on, only symmetric positive-definite matrices A will be considered. Let L and U be the respective lower and upper triangular Cholesky factors of A: $A = L^T L = U^T U$. Since the generalized Bareiss algorithm computes the UL and LU decompositions of A with unit triangular left factors, the algorithm could be applied to obtain the Cholesky factors L and U of A by simply scaling every row in the right factors by the square-root of the corresponding diagonal element.

3.1. Derivation of the Hyperbolic Cholesky Algorithm

A more balanced alternative, presented in the theorem below, is to modify the generalized Bareiss algorithm and scale intermediate quantities in order to obtain the Cholesky factors directly. Note that the theorem and its proof remain true for symmetric positive-definite matrices A with block entries if transposes are inserted in the right places.

Theorem 3.1. *If* $v_i^{(+0)} = v_i^{(-0)} = (a_{ii})^{-\frac{1}{2}} a_i$, $1 \le i \le n$, *and*

for $1 \le k \le n$, $1 \le i \le n-k$, $\rho_{i,i+k} = v^{(k-1)}_{i,i+k} (v^{(-k+1)}_{i+k,i+k})^{-1}$

$$\begin{pmatrix} v_i^{(k)} \\ v_{i+k}^{(-k)} \end{pmatrix} = (1 - \rho^2_{i,i+k})^{-\frac{1}{2}} \begin{pmatrix} 1 & -\rho_{i,i+k} \\ -\rho_{i,i+k} & 1 \end{pmatrix} \begin{pmatrix} v_i^{(k-1)} \\ v_{i+k}^{(-k+1)} \end{pmatrix},$$

then

$$a_i^{(k)} = (a_{ii}^{(k)})^{\frac{1}{2}} v_i^{(k)}, \qquad a_{i+k}^{(-k)} = (a^{(-k)}_{i+k,i+k})^{\frac{1}{2}} v^{(-k)}_{i+k}.$$

Proof. The proof is by induction. First note that since A is positive-definite a_{ii} is strictly positive, the initialization can be performed, and the theorem holds for $k = 0$. Suppose it also holds for $k \ge 0$. From the generalized Bareiss algorithm and the induction hypothesis

$$\begin{aligned} a_i^{(k+1)} &= a_i^{(k)} - a^{(k)}_{i,i+k+1} (a^{(-k)}_{i+k+1,i+k+1})^{-1} a^{(-k)}_{i+k+1} \\ &= (a_{ii}^{(k)})^{\frac{1}{2}} \left[v_i^{(k)} - (a_{ii}^{(k)})^{-\frac{1}{2}} a^{(k)}_{i,i+k+1} (a^{(-k)}_{i+k+1,i+k+1})^{-\frac{1}{2}} v^{(-k)}_{i+k+1} \right] \\ &= (a_{ii}^{(k)})^{\frac{1}{2}} \left[v_i^{(k)} - \rho_{i,i+k+1} v^{(-k)}_{i+k+1} \right]. \end{aligned}$$

Now, from the quotient property

$$\begin{aligned} a_{ii}^{(k+1)} &= a_{ii}^{(k)} - a^{(k)}_{i,i+k+1} (a^{(-k)}_{i+k+1,i+k+1})^{-1} a^{(-k)}_{i+k+1,i} \\ &= (a_{ii}^{(k)})^{\frac{1}{2}} (1 - \rho^2_{i,i+k+1}) (a_{ii}^{(k)})^{\frac{1}{2}}. \end{aligned}$$

Since A is positive-definite and, from the expression of the determinant of a contiguous principal submatrix as a product in Section 2.3, $a_{ii}^{(k+1)}$ is the ratio of the determinant of the $(k+2)$-block after row $i-1$ to the determinant of the $(k+1)$-block after row i it follows that $a_{ii}^{(k+1)} > 0$, $|\rho_{i,i+k+1}| < 1$, and

$$(a_{ii}^{(k)})^{\frac{1}{2}} = (a_{ii}^{(k+1)})^{\frac{1}{2}} (1 - \rho^2_{i,i+k+1})^{-\frac{1}{2}}.$$

Consequently,

$$a_i^{(k+1)} = (a_{ii}^{(k+1)})^{\frac{1}{2}} v_i^{(k+1)}.$$

The generalized Bareiss algorithm and the induction hypothesis also imply

$$\begin{aligned} a^{(-k-1)}_{i+k+1} &= a^{(-k)}_{i+k+1} - a^{(-k)}_{i+k+1,i} (a_{ii}^{(k)})^{-1} a_i^{(k)} \\ &= (a^{(-k)}_{i+k+1,i+k+1})^{\frac{1}{2}} \left[v^{(-k)}_{i+k+1} - (a^{(-k)}_{i+k+1,i+k+1})^{-\frac{1}{2}} a^{(-k)}_{i+k+1,i} (a_{ii}^{(-k)})^{-\frac{1}{2}} v_i^{(k)} \right] \\ &= (a^{(-k)}_{i+k+1,i+k+1})^{\frac{1}{2}} \left[v^{(-k)}_{i+k+1} - \rho_{i,i+k+1} v_i^{(k)} \right], \end{aligned}$$

where the last statement is obtained by observing that $a^{(-k)}_{i+k+1,i} = a^{(k)}_{i,i+k+1}$ for a symmetric matrix. Making again use of the quotient property

$$(a^{(-k)}_{i+k+1,i+k+1})^{\frac{1}{2}} = (a^{(-k-1)}_{i+k+1,i+k+1})^{\frac{1}{2}} (1 - \rho^2_{i,i+k+1})^{-\frac{1}{2}}.$$

Thus

$$a^{(-k-1)}_{i+k+1} = (a^{(-k-1)}_{i+k+1,i+k+1})^{\frac{1}{2}} v^{(-k-1)}_{i+k+1}.$$

∎

Since the operations applied to pairs of rows are hyperbolic rotations, the algorithm has been called the Hyperbolic Cholesky algorithm in [5]:

Hyperbolic Cholesky Algorithm

$$1 \le i \le n, \quad \begin{pmatrix} v_i^{(+0)} \\ v_i^{(-0)} \end{pmatrix} = \begin{pmatrix} (a_{ii})^{-\frac{1}{2}} a_i \\ (a_{ii})^{-\frac{1}{2}} a_i \end{pmatrix}$$

for $k = 1$ to $n-1$,

$$1 \le i \le n-k, \quad \rho_{i,i+k} = v_{i,i+k}^{(k-1)} (v_{i+k,i+k}^{(-k+1)})^{-1}$$

$$\begin{pmatrix} v_i^{(k)} \\ v_{i+k}^{(-k)} \end{pmatrix} = (1 - \rho_{i,i+k}^2)^{-\frac{1}{2}} \begin{pmatrix} 1 & -\rho_{i,i+k} \\ -\rho_{i,i+k} & 1 \end{pmatrix} \begin{pmatrix} v_i^{(k-1)} \\ v_{i+k}^{(-k+1)} \end{pmatrix}.$$

The rows of the Cholesky factors are given by

$$L = \begin{pmatrix} v_1^{(n-1)} \\ \vdots \\ v_{n-1}^{(1)} \\ v_n^{(0)} \end{pmatrix} = \begin{pmatrix} (a_{11}^{(n-1)})^{-\frac{1}{2}} a_1^{(n-1)} \\ \vdots \\ (a_{n-1,n-1}^{(1)})^{-\frac{1}{2}} a_{n-1}^{(1)} \\ (a_{nn}^{(0)})^{-\frac{1}{2}} a_n^{(0)} \end{pmatrix}, \qquad U = \begin{pmatrix} v_1^{(-0)} \\ v_2^{(-1)} \\ \vdots \\ v_n^{(-n+1)} \end{pmatrix} = \begin{pmatrix} (a_{11}^{(-0)})^{-\frac{1}{2}} a_1^{(-0)} \\ (a_{22}^{(-1)})^{-\frac{1}{2}} a_2^{(-1)} \\ \vdots \\ (a_{nn}^{(-n+1)})^{-\frac{1}{2}} a_n^{(-n+1)} \end{pmatrix}.$$

This algorithm differs slightly from the one presented in [5] because it computes both L and U factors. The algorithm in [5] determines only U (or only L), and takes advantage of the property that the 2×2 hyperbolic rotation matrices may be computed from the upper triangular (lower triangular) part of each row. This property follows from the symmetry of A; when applied to a non-symmetric matrix the generalized Bareiss algorithm cannot be decomposed into two such independent parts.

The operation count for computing both Cholesky factors is $\frac{1}{2}n(n-1)$ square roots, $n(n-1)$ divisions and about $\frac{4}{3}n^3$ multiplications; this is four times the number of operations of the Cholesky algorithm. However the number of hyperbolic rotations is the same as the number of multiplications in the Cholesky algorithm and, since specialized hardware may be devised that determines and applies hyperbolic rotations as fast (on about twice the area) as sequential multipliers, the Hyperbolic Cholesky algorithm may in some instances be competitive with the Cholesky algorithm. In the context of parallel computation on systolic arrays, with nearest neighbor processor interconnections and no broadcasting, Hyperbolic Cholesky computes a single Cholesky factor faster than the Cholesky algorithm. Hence the more complex processors bring about an advantage in speed [8].

3.2. Relation between Upper and Lower Triangular Cholesky Factor

With regard to parallel computation, data flow graphs have become a popular tool to visualize and analyze the partial order of computations in an algorithm. For instance, the graph in Figure 1 depicts the partial order of computations in the Hyperbolic Cholesky algorithm for $n = 4$; hyperbolic

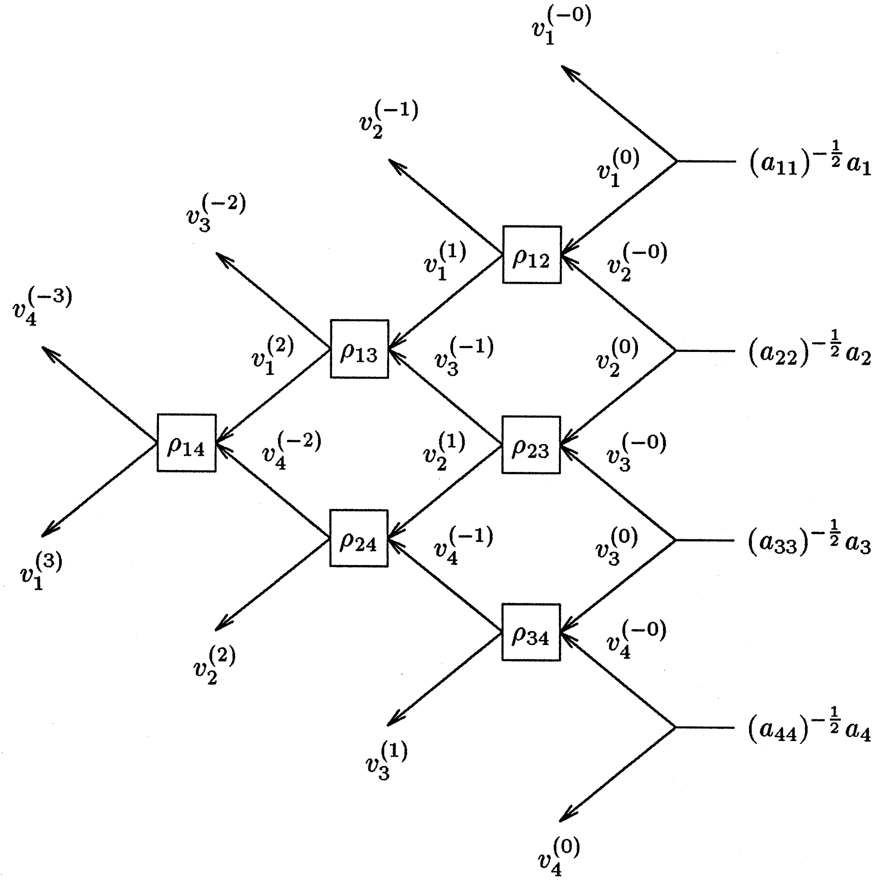

Figure 1: Data Flow Graph for the Hyperbolic Cholesky Algorithm, $n = 4$.

rotations whose tangents $\rho_{i,i+k}$ have the same index k belong to the same column, thus clearly showing that they are independent and can be determined in parallel.

The graph can also be viewed as the realization of a linear transformation in terms of elementary operations (hyperbolic rotations); the transformation takes as input from the right the scaled rows $(a_{ii})^{-\frac{1}{2}}a_i$ of A, and delivers the rows of U as outputs at the top and the rows of L as outputs at the bottom. Each box in Figure 1 can be viewed as a linear four-terminal element, with two inputs from the right and two outputs to the left. This interpretation illustrates the usefulness of the data flow graph not only for parallel algorithm analysis but also for parallel algorithm *synthesis*. Since U and L are Cholesky factors of the same matrix A they must be related by a $n \times n$ orthogonal transformation, and the graph can help us to discover a decomposition of this orthogonal transformation in terms of elementary, 2×2, orthogonal rotations. We simply exchange, for every element, the roles of the bottom right input and the top left output, which now become respectively output and input. The corresponding data flow graph is displayed in Figure 2; the effect of this exchange on the parameters of the four-terminal elements is made explicit in the theorem below. Note that the intermediate quantities on the edges in Figures 1 and 2 are identical.

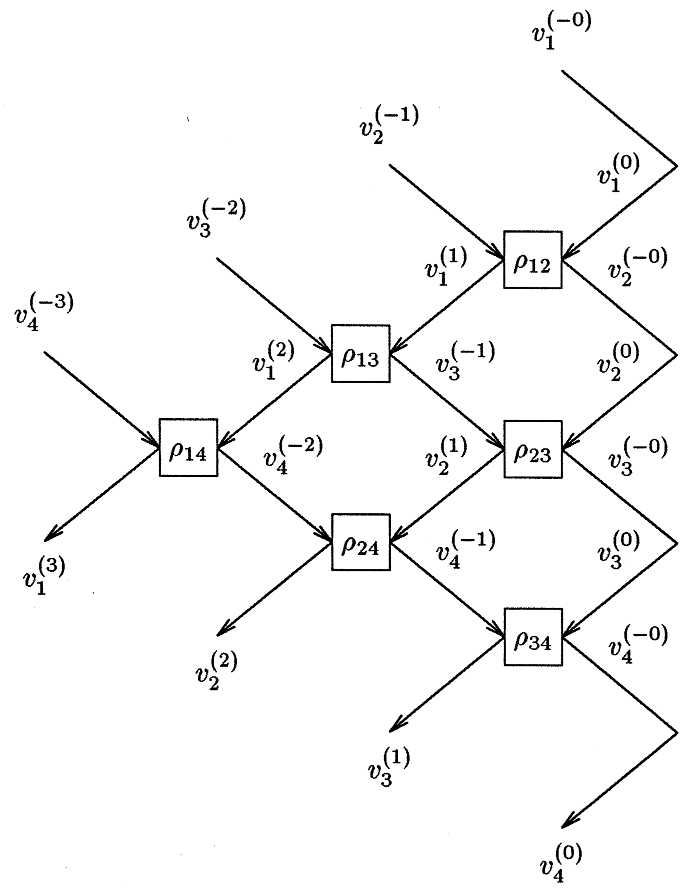

Figure 2: Data Flow Graph for the Cholesky Factor Interchange Algorithm, $n = 4$.

Theorem 3.2. *The following algorithm transforms the upper triangular Cholesky factor*

$$U = \begin{pmatrix} v_1^{(-0)} \\ v_2^{(-1)} \\ \vdots \\ v_n^{(-n+1)} \end{pmatrix}$$

of A to its lower triangular Cholesky factor L:

Cholesky Factor Interchange Algorithm

```
for $2i + k = 3$ to $2n - 1$,
    $1 \le i \le n,\ k \ge 1, \quad t_{i,i+k} = v_{i,i+k}^{(k-1)} (v_{i+k,i+k}^{(-k)})^{-1}$
        $\begin{pmatrix} v_i^{(k)} \\ v_{i+k}^{(-k+1)} \end{pmatrix} = (1 + t_{i,i+k}^2)^{-\frac{1}{2}} \begin{pmatrix} 1 & -t_{i,i+k} \\ t_{i,i+k} & 1 \end{pmatrix} \begin{pmatrix} v_i^{(k-1)} \\ v_{i+k}^{(-k)} \end{pmatrix}.$
```

Proof. If two vectors $(\, v' \quad w' \,)^T$ and $(\, v \quad w \,)^T$ are related by a hyperbolic rotation

$$\begin{pmatrix} v' \\ w' \end{pmatrix} = \begin{pmatrix} c_h & -s_h \\ -s_h & c_h \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}, \qquad c_h^2 - s_h^2 = 1$$

then the vectors $(v' \quad w)^T$ and $(v \quad w')^T$ are related by

$$\begin{pmatrix} v' \\ w \end{pmatrix} = \begin{pmatrix} c_h^{-1} & -t_h \\ t_h & c_h^{-1} \end{pmatrix} \begin{pmatrix} v \\ w' \end{pmatrix}, \qquad t_h = s_h / c_h.$$

Since $c_h^{-1} = (1 - t_h^2)^{\frac{1}{2}}$, the 2×2 matrix above is an orthogonal rotation

$$\begin{pmatrix} v' \\ w \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} v \\ w' \end{pmatrix}, \qquad s = t_h, \quad c = (1 - s^2)^{\frac{1}{2}}.$$

Thus, if we have

$$\begin{pmatrix} v_i^{(k)} \\ v_{i+k}^{(-k)} \end{pmatrix} = (1 - \rho_{i,i+k}^2)^{-\frac{1}{2}} \begin{pmatrix} 1 & -\rho_{i,i+k} \\ -\rho_{i,i+k} & 1 \end{pmatrix} \begin{pmatrix} v_i^{(k-1)} \\ v_{i+k}^{(-k+1)} \end{pmatrix}$$

from the Hyperbolic Cholesky algorithm then

$$\begin{pmatrix} v_i^{(k)} \\ v_{i+k}^{(-k+1)} \end{pmatrix} = \begin{pmatrix} c_{i,i+k} & -s_{i,i+k} \\ s_{i,i+k} & c_{i,i+k} \end{pmatrix} \begin{pmatrix} v_i^{(k-1)} \\ v_{i+k}^{(-k)} \end{pmatrix}$$

where the elements of the rotation satisfy $s_{i,i+k} = \rho_{i,i+k}$ and $c_{i,i+k} = (1 - \rho_{i,i+k}^2)^{\frac{1}{2}}$. This relation does not yet provide us with the complete algorithm: $\rho_{i,i+k}$ is equal to $v_{i,i+k}^{(k-1)}(v_{i+k,i+k}^{(-k+1)})^{-1}$ but this relationship cannot be employed directly since $v_{i+k,i+k}^{(-k+1)}$ is not available. However, the Schur complement interpretation of $v_i^{(k)}$ indicates that $v_{i,i+k}^{(k)} = 0$ hence the rotation satisfies

$$\begin{pmatrix} c_{i,i+k} & -s_{i,i+k} \\ s_{i,i+k} & c_{i,i+k} \end{pmatrix} \begin{pmatrix} v_{i,i+k}^{(k-1)} \\ v_{i+k,i+k}^{(-k)} \end{pmatrix} = \begin{pmatrix} 0 \\ v_{i+k,i+k}^{(-k+1)} \end{pmatrix}$$

from which it directly follows that $c_{i,i+k} = (1 + t_{i,i+k}^2)^{-\frac{1}{2}}$ and $s_{i,i+k} = c_{i,i+k} t_{i,i+k}$ with $t_{i,i+k} = v_{i,i+k}^{(k-1)}(v_{i+k,i+k}^{(-k)})^{-1}$. Since at each step the superscript of every row being updated is incremented by 1, the rotations can clearly be executed in the specified order and the result matrix equals

$$\begin{pmatrix} v_1^{(n-1)} \\ \vdots \\ v_{n-1}^{(1)} \\ v_n^{(0)} \end{pmatrix}.$$

But this is just the lower triangular Cholesky factor L of A as produced by the Hyperbolic Cholesky algorithm.

▮

4. Application to the Computation of Sample Partial Correlations

Now consider the case where the $n \times n$ symmetric positive-definite matrix A represents the sample covariance matrix of a $m \times n$ zero-mean data matrix B: $A = (m-1)^{-1} B^T B$ [1] (without loss of generality, the scaling factor $(m-1)^{-1}$ is omitted in the sequel). The $m \times n$ matrix

$$B = (b_1 \quad \dots \quad b_n)$$

contains a zero-mean column vector b_i for each of n real random variables B_i, $1 \leq i \leq n$. Now the results of the previous section are employed to derive an algorithm for the computation of certain sample partial correlations of B that does not require formation of $B^T B$. We will give both, algebraic and geometric derivations of the algorithm.

4.1. Algebraic Derivation

From Theorem 3.1 we know that $\rho_{i,i+1}$ satisfies

$$\rho_{i,i+1} = v^{(0)}_{i,i+1}(v^{(-0)}_{i+1,i+1})^{-1} = (a_{ii})^{-\frac{1}{2}} a_{i,i+1}(a_{i+1,i+1})^{-\frac{1}{2}}, \qquad 1 \leq i \leq n-1.$$

Since $A = B^T B$, $\rho_{i,i+1}$ represents the *sample correlation coefficient between B_i and B_{i+1}* [1]. Furthermore,

$$\rho_{i,i+2} = v^{(1)}_{i,i+2}(v^{(-1)}_{i+2,i+2})^{-1} = (a^{(1)}_{ii})^{-\frac{1}{2}} a^{(1)}_{i,i+2}(a^{(-1)}_{i+2,i+2})^{-\frac{1}{2}}, \qquad 1 \leq i \leq n-2.$$

Hence, $\rho_{i,i+2}$ represents the *sample partial correlation coefficient between B_i and B_{i+2} holding B_{i+1} fixed* [1]. In general,

$$\rho_{i,i+k} = v^{(k-1)}_{i,i+k}(v^{(-k+1)}_{i+k,i+k})^{-1} = (a^{(k-1)}_{ii})^{-\frac{1}{2}} a^{(k-1)}_{i,i+k}(a^{(-k+1)}_{i+k,i+k})^{-\frac{1}{2}}, \qquad 1 \leq k \leq n-i,$$

is the sample partial correlation coefficient between B_i and B_{i+k} holding the variables inbetween, $B_{i+1} \ldots B_{i+k-1}$, fixed [1].

The upper triangular matrix U in the QR factorization $B = QU$ of B (where Q has orthonormal columns) is also the upper triangular Cholesky factor of A since $A = U^T Q^T Q U = U^T U$. Theorem 3.2 asserts that, in the transformation from U to L, the sines of the rotations are equal to the partial correlations.

Accordingly, the following algorithm computes the partial correlations $\rho_{i,i+k}$ directly from the data matrix B:

1. Compute the QR decomposition $B = QU$ of B, where Q has orthonormal columns and U is upper triangular with positive diagonal elements.
2. Use the Cholesky Factor Interchange algorithm to transform U to L.
3. The sine of the rotation eliminating element $(i, i+k)$ is the sample partial correlation coefficient $\rho_{i,i+k}$ between B_i and B_{i+k}, holding the variables inbetween, $B_{i+1} \ldots B_{i+k-1}$, fixed.

As shown in [6, 7] this algorithm avoids the loss of numerical accuracy associated with the formation of $B^T B$.

4.2. Geometric Derivation

Because a purely algebraic interpretation does not seem to be very insightful, we offer a geometric derivation of Theorem 3.2, starting with two simple examples.

First consider the 2×2 case. Let

$$U = \begin{pmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{pmatrix}$$

be the upper triangular matrix (with positive diagonal elements) in the QR decomposition of the $m \times 2$ matrix B. Since $a_{ij} = b_i^T b_j$ we know from Theorem 3.1 that the sample correlation coefficient equals

$$\rho_{i,i+1} = (b_i^T b_i)^{-\frac{1}{2}}(b_i^T b_{i+1})(b_{i+1}^T b_{i+1})^{-\frac{1}{2}}$$

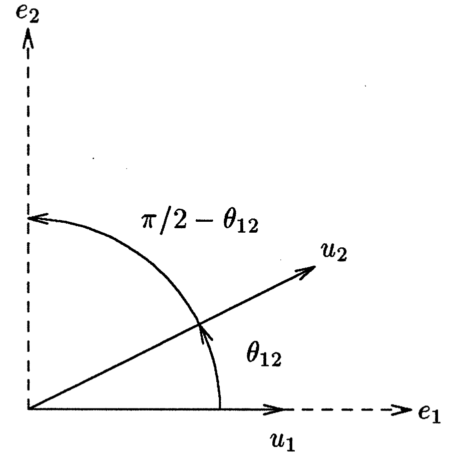

Figure 3: Angles in the 2 × 2 Example.

and thus represents the cosine of the angle between B_i and B_{i+1}. So, the sample correlation ρ_{12} between B_1 and B_2 is the cosine of the angle θ_{12} between the two columns of U. Because of the triangular structure of U its first column, $(\,u_{11} \quad 0\,)^T$, is a positive multiple of the first canonical vector $e_1 = (\,1 \quad 0\,)^T$ while its second column is a linear combination of e_1 and the second canonical vector $e_2 = (\,0 \quad 1\,)^T$.

The columns of the matrix U may be rotated in such a way that the second column becomes a positive multiple of e_2 thereby turning the first into a linear combination of e_1 and e_2:

$$L \equiv \Theta U = \begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix}.$$

The angle between e_1 and e_2, denoted by $\angle(e_1, e_2)$, is $+\pi/2$ and, using the same orientation convention, the angle between the two columns is denoted by $\theta_{12} \equiv \angle(u_1, u_2)$. The fact that the first column is a positive multiple of e_1 implies $\angle(e_1, u_2) = \theta_{12}$. To turn the second column into a positive multiple of e_2 requires that all columns of U be rotated by the angle

$$\angle(u_2, e_2) = \angle(e_1, e_2) - \angle(e_1, u_2) = \pi/2 - \theta_{12},$$

see Figure 3. Since the angle between the two columns of U is preserved under the rotation, and the angle of such a rotation

$$\Theta = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

completes θ_{12} to a right angle:

$$s = \sin(\pi/2 - \theta_{12}) = \cos\theta_{12} = \rho_{12}.$$

Consequently, the desired sample correlation is the sine of the rotation Θ.

Let us take a brief look at the 3 × 3 case

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

At first, because the second column has only one more non-zero element than the first, the columns of U can be rotated in the (e_1, e_2)-plane so as to make the second column co-linear with e_2,

$$\begin{pmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = \begin{pmatrix} * & 0 & * \\ * & * & * \\ 0 & 0 & u_{33} \end{pmatrix},$$

and $\rho_{12} = s_{12}$. Here, 'co-linear' is used to mean 'a positive multiple of' and $*$ denotes terms that are non-zero in general.

In order to provide the geometric meaning of the *partial* correlations we appeal again to Theorem 3.1:

$$\rho_{i,i+2} = v^{(1)}_{i,i+2}(v^{(-1)}_{i+2,i+2})^{-1} = (a^{(1)}_{ii})^{-\frac{1}{2}} a^{(1)}_{i,i+2}(a^{(-1)}_{i+2,i+2})^{-\frac{1}{2}}, \qquad 1 \le i \le n-2,$$

where

$$\begin{aligned} a^{(1)}_{i,i+2} &= a_{i,i+2} - a_{i,i+1}(a_{i+1,i+1})^{-1}a_{i+1,i+2} = (b_i^T b_{i+2}) - (b_i^T b_{i+1})(b_{i+1}^T b_{i+1})^{-1}(b_{i+1}^T b_{i+2}) \\ &= b_i^T[I_m - b_{i+1}(b_{i+1}^T b_{i+1})^{-1}b_{i+1}^T]b_{i+2}. \end{aligned}$$

The center matrix

$$I_m - b_{i+1}(b_{i+1}^T b_{i+1})^{-1}b_{i+1}^T$$

is an orthogonal projector onto the subspace orthogonal to b_{i+1} and as such symmetric and idempotent, so that

$$a^{(1)}_{i,i+2} = b_i^T(I_m - b_{i+1}(b_{i+1}^T b_{i+1})^{-1}b_{i+1}^T)^T(I_m - b_{i+1}(b_{i+1}^T b_{i+1})^{-1}b_{i+1}^T)b_{i+2} = b_i^{(1)}b_{i+2}^{(-1)},$$

where

$$b_i^{(1)} \equiv b_i - b_{i+1}(b_{i+1}^T b_{i+1})^{-1}(b_{i+1}^T b_i), \qquad b_{i+2}^{(-1)} \equiv b_{i+2} - b_{i+1}(b_{i+1}^T b_{i+1})^{-1}(b_{i+1}^T b_{i+2})$$

are the respective projections of b_i and b_{i+2} onto the subspace orthogonal to the vector inbetween: b_{i+1}. Similarly,

$$a^{(1)}_{ii} \equiv (b_i^{(1)})^T b_i^{(1)}, \qquad a^{(-1)}_{i+2,i+2} \equiv (b_{i+2}^{(-1)})^T b_{i+2}^{(-1)}.$$

Hence,

$$\rho_{i,i+2} = [(b_i^{(1)})^T b_i^{(1)}]^{-\frac{1}{2}}[(b_i^{(1)})^T b_{i+2}^{(-1)}][(b_{i+2}^{(-1)})^T b_{i+2}^{(-1)}]^{-\frac{1}{2}}$$

represents the cosine of the angle between the respective projections of b_i and b_{i+2} onto the subspace orthogonal to the vector inbetween, b_{i+1}.

Thus, to achieve conditioning of B_1 and B_3 with respect to B_2, the first and third columns need to be projected onto the subspace orthogonal to the second column. Due to the triangular structure of U and the effect of the previous rotation the second column is co-linear to e_2, and the subspace orthogonal to it is just the plane (e_1, e_3). The partial correlation ρ_{13} can then be determined from that rotation that makes co-linear with e_3 the projection of the third column onto (e_1, e_3). Since this rotation takes place in a subspace orthogonal to the second column it does not affect the second column, and the zero element introduced by the previous rotation is preserved:

$$\begin{pmatrix} c_{13} & 0 & -s_{13} \\ 0 & 1 & 0 \\ s_{13} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} * & 0 & * \\ * & * & * \\ 0 & 0 & u_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ * & * & * \\ * & 0 & * \end{pmatrix}$$

and $\rho_{13} = s_{13}$. Note that another non-zero element is introduced in the first column.

Again, because of the triangular structure of U and the effect of the second rotation the zero-structure of the second and third columns is the same save for one element, the second column is co-linear with e_2 while the third is a linear combination of e_2 and e_3. Thus the columns of the matrix can be rotated to yield ρ_{23} by applying a rotation that makes the whole third column co-linear with e_3, and turns the second column into a linear combination of e_2 and e_3.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & -s_{23} \\ 0 & s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} l_{11} & 0 & 0 \\ * & * & * \\ * & 0 & * \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix}$$

and $\rho_{23} = s_{23}$.

Now we are ready to carry out this geometric argument in the general case. As it is easier for the induction proof to follow a sequential order of elimination the version of the algorithm presented in the theorem below is sequential.

Theorem 4.1. *Geometric Version of Theorem 3.2.*
If the elements in the Cholesky factor U of the sample covariance matrix A are eliminated in the order

$$\begin{pmatrix} * & 1 & 2 & \dots & n-1 \\ & * & n & & 2n-3 \\ & & \ddots & & \vdots \\ & & & * & n(n-1)/2 \\ & & & & * \end{pmatrix},$$

that is, proceeding row after row from top to bottom, and within each row from left to right, then the sine of the rotation that eliminates element $(i, i+k)$, is equal to the sample partial correlation $\rho_{i,i+k}$, $1 \le i \le n-1$, $1 \le k \le n-i$.

Proof. The proof proceeds by induction.

The induction basis comprises the computation of partial correlations between B_1 and all other variables. To start with, the matrix U is of the form

$$\begin{pmatrix} * & * & * & * & \dots & * \\ & * & * & * & \dots & * \\ & & * & * & \dots & * \\ & & & * & \dots & * \\ & & & & \ddots & \vdots \\ & & & & & * \end{pmatrix}.$$

From the 2×2 case one can see that elimination of element $(1, 2)$ in the upper triangular matrix U by a rotation in plane (e_1, e_2) provides ρ_{12}. The second column of the resulting matrix becomes co-linear to e_2 while the first column becomes a linear combination of e_1 and e_2. Hence, there is a new non-zero element in the first column and a zero has been introduced in the first row:

$$\begin{pmatrix} * & & * & * & \dots & * \\ * & * & * & * & \dots & * \\ & & * & * & \dots & * \\ & & & * & \dots & * \\ & & & & \ddots & \vdots \\ & & & & & * \end{pmatrix}.$$

The 3×3 case showed that the correlation ρ_{13} between B_1 and B_3 given B_2 could be computed by rotating the first and third column and thereby introducing a non-zero element in position $(3,1)$ and a zero in position $(1,3)$:

$$\begin{pmatrix} * & & & * & \cdots & * \\ * & * & * & * & \cdots & * \\ * & & * & * & \cdots & * \\ & & & * & \cdots & * \\ & & & & \ddots & \vdots \\ & & & & & * \end{pmatrix}.$$

Continuing this argument, the partial correlation $\rho_{1,1+k}$ between B_1 and B_{1+k}, given B_2, ..., B_k is computed by performing a rotation in plane (e_1, e_{1+k}) thereby creating a zero element in position $(1, 1+k)$, $1 \leq k \leq n-1$. Thus, once all correlations involving B_1 have been computed the first column of the matrix has totally filled in, and the first row is zero except for the first element:

$$\begin{pmatrix} * & & & & & \\ * & * & * & * & \cdots & * \\ * & & * & * & \cdots & * \\ * & & & * & \cdots & * \\ \vdots & & & & \ddots & \vdots \\ * & & & & & * \end{pmatrix}.$$

Assume that the partial correlations $\rho_{i,i+k-1}$ have already been computed for $1 \leq i$, $1 \leq k-1 < n-i$. The corresponding matrix is of the form

$$\left[\begin{array}{c|ccccc|c} L_0 & & & & & & \\ \hline * & w_{ii} & & & & w_{i,i+k} & * \\ * & w_{i+1,i} & w_{i+1,i+1} & \cdots & w_{i+1,i+k-1} & w_{i+1,i+k} & * \\ * & \vdots & & \ddots & \vdots & \vdots & * \\ * & w_{i+k-1,i} & & & w_{i+k-1,i+k-1} & w_{i+k-1,i+k} & * \\ * & & & & & w_{i+k,i+k} & * \\ \hline * & & & & & & U_0 \end{array}\right],$$

where L_0 is lower triangular and U_0 is upper triangular.

By induction hypothesis the entire lower triangular part of the leading $i-1$ columns is non-zero, and the ith column has $k-1$ non-zeros in its lower triangular part due to the computation of $\rho_{i,i+1}$, ..., $\rho_{i,i+k-1}$.

In order to compute the next correlation $\rho_{i,i+k}$ the corresponding columns w_i, ..., w_{i+k} of the current matrix must be projected onto a subspace orthogonal to the subspace spanned by B_{i+1}, ..., B_{i+k-1} (by an induction on k along the same lines as above one can easily show that $\rho_{i,i+k}$ represents the cosine of the angle between the respective projections of b_i and b_{i+k} onto the subspace orthogonal to the vectors inbetween, $b_{i+1} \ldots b_{i+k-1}$). Due to the initial 'nesting' of the column subspaces (i.e. the original upper triangular structure of U) the trailing components $i+k$, ..., n of w_{i+1}, ..., w_{i+k-1} are zero; and due to the rotations performed in order to retrieve previous partial correlations (i.e. the appearing lower triangular structure of L) the leading components 1, ..., i of w_{i+1}, ..., w_{i+k-1} are zero. Hence the subspace spanned by B_{i+1}, ..., B_{i+k-1} is the space spanned by e_{i+1}, ..., e_{i+k-1}, and the space orthogonal to it is the space spanned by e_1, ..., e_i, e_{i+k}, ..., e_n. Similarly, components 1, ..., $i-1$, $i+k$, ..., n of w_i and components 1, ..., $i-1$, $i+k+1$, ..., n

of w_{i+k} are zero; and the projections of w_i and w_{i+k} onto $e_1, \ldots, e_i, e_{i+k}, \ldots, e_n$ are respectively co-linear to e_i and a linear combination of e_i and e_{i+k}. Thus, $\rho_{i,i+k}$ is obtained by applying the rotation in plane (e_i, e_{i+k}) that makes the projection of w_{i+k} co-linear with e_i; $\rho_{i,i+k}$ is the sine of that rotation. After the rotation the matrix has the form

$$\left[\begin{array}{c|ccccc|c}
L_0 & & & & & & \\ \hline
* & w'_{ii} & & & & & * \\
* & w_{i+1,i} & w_{i+1,i+1} & \cdots & w_{i+1,i+k-1} & w_{i+1,i+k} & * \\
* & \vdots & & \ddots & \vdots & \vdots & * \\
* & w_{i+k-1,i} & & & w_{i+k-1,i+k-1} & w_{i+k-1,i+k} & * \\
* & w'_{i+k,i} & & & & w'_{i+k,i+k} & * \\ \hline
* & & & & & & U_0
\end{array}\right].$$

$\blacksquare$

According to Theorem 3.2, the elimination can be carried out in parallel as follows

$$\begin{pmatrix}
* & 1 & 2 & 3 & \ldots & n-2 & n-1 \\
 & * & 3 & 4 & \ldots & n-1 & n \\
 & & * & 5 & \ldots & n & n+1 \\
 & & & \ddots & & & \vdots \\
 & & & & \ddots & & \vdots \\
 & & & & & * & 2n-3 \\
 & & & & & & *
\end{pmatrix}.$$

If only the partial correlations are of interest, and the matrix L is not needed, about half of the arithmetic operations can be saved by applying the rotations merely to the trailing principal submatrix of interest.

5. Computation of Arbitrary Partial Correlations

Subject to a certain inital ordering of the random variables $B_1, \ldots, B_n$ our algorithm computes the partial correlations $\rho_{i,i+k}$ between B_i and B_{i+k} given $B_{i+1}, \ldots, B_{i+k-1}$ by completely reducing the upper triangular matrix U to a lower triangular matrix L. However, one may want to compute partial correlations where variables other than the ones inbetween are held fixed. To this end, we extend our notation by introducing superscripts for the partial correlations to indicate the fixed variables: the old $\rho_{i,i+k}$ is replaced by the new $\rho_{i,i+k}^{i+1:i+k-1}$, and in general $\rho_{i,j}^{S}$ denotes the partial correlation between variables B_i and B_j with variables held fixed whose indices belong to S $(i,\ j \notin S)$.

Now, other partial correlations may be computed by performing only a partial reduction. For instance, consider the following 6×6 example

$$U = \begin{pmatrix}
u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\
 & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} \\
 & & u_{33} & u_{34} & u_{35} & u_{36} \\
 & & & u_{44} & u_{45} & u_{46} \\
 & & & & u_{55} & u_{56} \\
 & & & & & u_{66}
\end{pmatrix}.$$

The leading three columns of U span the subspace of B_1, B_2 and B_3, and this is equal to the space spanned by the first three canonical vectors e_1, e_2 and e_3 due to the triangular structure of U. The space orthogonal to it is the one spanned by e_4, e_5, e_6 and is, because of the triangular structure, equal to that of columns 4 through 6 of U with components 1 to 3 set to zero. This means that the correlation $\rho_{45}^{1:3}$ between B_4 and B_5, given B_1, B_2 and B_3, can be computed by a rotation of U in plane (e_4, e_5). The resulting matrix has a new zero in column five and a fill-in in column four:

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} \\ & & u_{33} & u_{34} & u_{35} & u_{36} \\ & & & * & & * \\ & & & * & * & * \\ & & & & & u_{66} \end{pmatrix}.$$

The next correlation that can be computed is $\rho_{46}^{1:3,5}$ with a rotation in plane (e_4, e_6), the subspace orthogonal to B_1, B_2, B_3 and B_5:

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} \\ & & u_{33} & u_{34} & u_{35} & u_{36} \\ & & & l'_{11} & & \\ & & & * & * & * \\ & & & * & & * \end{pmatrix}.$$

The last correlation $\rho_{56}^{1:3}$ is determined by completing the transformation of the 3×3 trailing principal submatrix to lower triangular form:

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} \\ & & u_{33} & u_{34} & u_{35} & u_{36} \\ & & & l'_{11} & & \\ & & & l'_{21} & l'_{22} & \\ & & & l'_{31} & l'_{32} & l'_{33} \end{pmatrix}.$$

In general, the correlation $\rho_{ij}^{1:\alpha,i+1:j-1}$ for $i > \alpha$ and $i < j \leq n$ can be determined be preserving the leading α rows and columns of U and transforming the trailing principal submatrix of order $n-\alpha$ to lower triangular form L_α by appropriate plane rotations:

$$\begin{pmatrix} * & \cdots & \cdots & \cdots & \cdots & * \\ & \ddots & & & & \vdots \\ & & * & \cdots & \cdots & * \\ & & & * & \cdots & * \\ & & & & \ddots & \vdots \\ & & & & & * \end{pmatrix} \to \begin{pmatrix} * & \cdots & \cdots & \cdots & \cdots & * \\ & \ddots & & & & \vdots \\ & & * & \cdots & \cdots & * \\ & & & * & & \\ & & & \vdots & \ddots & \\ & & & * & \cdots & * \end{pmatrix}.$$

Similarly, the computation of $\rho_{ij}^{i+1:j-1,n-\beta+1:n}$ for $j < n - \beta + 1$ and $1 \leq i < j$ is accomplished by transforming U to lower triangular form L (or obtaining directly a QL factorization of B) and then transforming the leading $\beta \times \beta$ principal submatrix of L to upper triangular form U_β:

$$\begin{pmatrix} * & \cdots & \cdots & \cdots & \cdots & * \\ & \ddots & & & & \vdots \\ & & * & \cdots & \cdots & * \\ & & & * & \cdots & * \\ & & & & \ddots & \vdots \\ & & & & & * \end{pmatrix} \to \begin{pmatrix} * & & & & & \\ \vdots & \ddots & & & & \\ * & \cdots & * & & & \\ * & \cdots & \cdots & * & & \\ \vdots & & & & \ddots & \\ * & \cdots & \cdots & \cdots & \cdots & * \end{pmatrix} \to \begin{pmatrix} * & \cdots & * & & & \\ & \ddots & \vdots & & & \\ & & * & & & \\ * & \cdots & \cdots & * & & \\ \vdots & & & & \ddots & \\ * & \cdots & \cdots & \cdots & \cdots & * \end{pmatrix}.$$

Combining the two above strategies makes it possible to determine $\rho_{ij}^{1:\alpha,i+1:j-1,n-\beta+1:n}$ for $\alpha < i < j < n - \beta + 1$ by transforming the trailing $(n-\alpha) \times (n-\alpha)$ principal submatrix of U to block lower triangular form L_α (see the sketch below) and subsequently transforming the leading $(\beta - \alpha) \times (\beta - \alpha)$ triangular submatrix of L_α to upper triangular form $U_{\alpha,\beta}$:

$$\begin{pmatrix}
* & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & * \\
 & \ddots & & & & & & & \vdots \\
 & & * & \cdots & \cdots & \cdots & \cdots & \cdots & * \\
 & & & * & \cdots & \cdots & \cdots & \cdots & * \\
 & & & & \ddots & & & & \vdots \\
 & & & & & * & \cdots & \cdots & * \\
 & & & & & & * & \cdots & * \\
 & & & & & & & \ddots & \vdots \\
 & & & & & & & & *
\end{pmatrix}
\longrightarrow
\begin{pmatrix}
* & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & * \\
 & \ddots & & & & & & & \vdots \\
 & & * & \cdots & \cdots & \cdots & \cdots & \cdots & * \\
 & & & * & & & & & \\
 & & & \vdots & \ddots & & & & \\
 & & & * & \cdots & * & & & \\
 & & & * & \cdots & \cdots & \cdots & \cdots & * \\
 & & & \vdots & & & & & \vdots \\
 & & & * & \cdots & \cdots & \cdots & \cdots & *
\end{pmatrix}$$

$$\longrightarrow
\begin{pmatrix}
* & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & * \\
 & \ddots & & & & & & & \vdots \\
 & & * & \cdots & \cdots & \cdots & \cdots & \cdots & * \\
 & & & * & \cdots & * & & & \\
 & & & & \ddots & \vdots & & & \\
 & & & & & * & & & \\
 & & & * & \cdots & \cdots & \cdots & \cdots & * \\
 & & & \vdots & & & & & \vdots \\
 & & & * & \cdots & \cdots & \cdots & \cdots & *
\end{pmatrix}$$

If it is known in advance which partial correlations are to be determined then the columns of the $m \times n$ data matrix may be ordered so as to minimize the number of arithmetic operations succeeding the computation of the Cholesky factor.

For instance, a lower bound on the number of arithmetic operations in the computation of ρ_{ij}^S, where S is a subset of $k \geq 0$ numbers in $1 \ \ldots \ n$ not containing i and j, is $O(n-k)$ since our method requires at least one rotation to compute a partial correlation and the dimension of the space involved is $n - k$. This lower bound is attained by ordering the columns so that the set S represents the leading k columns of the data matrix followed by columns B_i and B_j. The correlation ρ_{ij}^k can then be determined by one rotation in the plane (e_{k+1}, e_{k+2}) that, due to the triangular structure of the Cholesky factor, involves $O(n-k)$ non-zero element pairs.

Not only the ordering of the columns is important but also the sequence in which particular correlations are computed. Consider the computation of a partial correlation between two variables B_i and B_j with successively more variables fixed: $\rho_{ij}^{S_1}, \ldots, \rho_{ij}^{S_k}$, where $S_1 \subset \ldots \subset S_k$ and $i, j \not\in S_k$. It seems that the following order of rotations constitutes the simplest way of determining the above correlations. It is illustrated by means of a 5×5 example for the computation of ρ_{12}, ρ_{12}^3, $\rho_{12}^{3:4}$, and $\rho_{12}^{3:5}$. At first the columns of the data matrix are ordered so that i and j represent the first two columns followed by the columns of S_1, the columns of $S_2 - S_1$, the columns of $S_3 - S_2 - S_1$, etc. In the example this amounts to the 'natural' ordering $B_1 \ldots B_5$ of the variables. The first correlation ρ_{12} can now be computed with one rotation from the Cholesky factor U. To compute ρ_{12}^3 columns 2 and 3 of U are exchanged and a rotation in plane (e_2, e_3) results in the Cholesky factor U' corresponding to the data matrix with variables in the order B_1, B_3, B_2, B_4, B_5. Since B_3 is situated between B_1 and B_2 two rotations suffice for the computation of ρ_{12}^3. The effect of these steps on the matrix is depicted below:

$$\begin{pmatrix}
u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\
 & u_{22} & u_{23} & u_{24} & u_{25} \\
 & & u_{33} & u_{34} & u_{35} \\
 & & & u_{44} & u_{45} \\
 & & & & u_{55}
\end{pmatrix}
\longrightarrow
\begin{pmatrix}
u_{11} & u_{13} & u_{12} & u_{14} & u_{15} \\
 & u_{23} & u_{22} & u_{24} & u_{25} \\
 & u_{33} & & u_{34} & u_{35} \\
 & & & u_{44} & u_{45} \\
 & & & & u_{55}
\end{pmatrix}
\longrightarrow
\begin{pmatrix}
u_{11} & u_{13} & u_{12} & u_{14} & u_{15} \\
 & u'_{22} & u'_{23} & u'_{24} & u'_{25} \\
 & & u'_{33} & u'_{34} & u'_{35} \\
 & & & u_{44} & u_{45} \\
 & & & & u_{55}
\end{pmatrix}$$

$$\rightarrow \begin{pmatrix} * & & * & * & * \\ * & * & * & * & * \\ & & u'_{33} & u'_{34} & u'_{35} \\ & & & u_{44} & u_{45} \\ & & & & u_{55} \end{pmatrix} \rightarrow \begin{pmatrix} * & & & * & * \\ * & * & * & * & * \\ * & & * & * & * \\ & & & u_{44} & u_{45} \\ & & & & u_{55} \end{pmatrix}.$$

Similarly, exchanging columns 3 and 4 of U', performing a rotation in plane (e_3, e_4) to get the Cholesky factor U'' of the data matrix corresponding to the ordering B_1, B_3, B_4, B_2, B_5, and performing three more rotations on U'' results in the extraction of $\rho_{12}^{3:4}$. In general, if the sets S_l differ by more than one index, more columns of the Cholesky factor must be exchanged to ensure that all fixed variables are situated between B_i and B_j.

As for arbitrary sequences of partial correlations, the determination of the column ordering of the data matrix as well as the computation sequence of the partial correlations so as to minimize the number of arithmetic operations seems to be an NP-complete problem. The use of heuristics, such as the following greedy approach, might lead to acceptable operation counts: the random variables are ordered so that as many partial correlations as possible can be determined from the resulting Cholesky factor. Repeatedly, the columns of the Cholesky factor are then re-ordered according to the same strategy, the matrix returned to upper triangular form, and appropriate rotations performed until all correlations have been computed.

References

[1] Anderson, T.W., *An Introduction to Multivariate Statistical Analysis, Second Edition,* Wiley, 1984.

[2] Ando, T., *Generalized Schur Complements,* Linear Algebra and its Applications, 27 (1979), pp. 173–86.

[3] Bareiss, E.H., *Numerical Solution of Linear Equations with Toeplitz and Vector Toeplitz Matrices,* Numer. Math., 13 (1969), pp. 404–24.

[4] Cottle, R.W., *Manifestations of the Schur Complement,* Linear Algebra and its Applications, 8 (1974), pp. 189–211.

[5] Delosme, J.-M. and Ipsen, I.C.F., *Parallel Solution of Symmetric Positive Definite Systems with Hyperbolic Rotations,* Linear Algebra and its Applications, 77 (1986), pp. 75–111.

[6] ————, *Computing Partial Correlations from the Data Matrix,* Research Report 541, Dept Computer Science, Yale University, 1987.

[7] Delosme, J.-M., Ipsen, I.C.F. and Paige, C.C, *The Cholesky Factorization, Schur Complements, Correlation Coefficients, Angles between Vectors, and the QR Factorization,* Research Report 607, Dept Computer Science, Yale University, 1988.

[8] Delosme, J.-M. and Morf, M., *Covariance Decompositions Via Elementary Transformations. Applications to Filtering,* Circuits, Systems and Signal Processing, 7 (1988), pp. 21–55.