



## Some Algorithms for the Conditional Mean Vector and Covariance Matrix

John F. Monahan

North Carolina State University

---

### Abstract

We consider here the problem of computing the mean vector and covariance matrix for a conditional normal distribution, considering especially a sequence of problems where the conditioning variables are changing. The sweep operator provides one simple general approach that is easy to implement and update. A second, more goal-oriented general method avoids explicit computation of the vector and matrix, while enabling easy evaluation of the conditional density for likelihood computation or easy generation from the conditional distribution. The covariance structure that arises from the special case of an ARMA( $p, q$ ) time series can be exploited for substantial improvements in computational efficiency.

*Keywords:* conditional distribution, sweep operator, ARMA process.

---

### 1. Introduction

Consider  $\mathbf{Z} \sim \text{Normal}_n(\boldsymbol{\mu}, \mathbf{A})$  and partition  $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2)^\top$ , similarly,  $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)^\top$ , and the covariance matrix  $\mathbf{A}$ , to write

$$\begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix} \sim N_n \left( \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \right) \quad \begin{matrix} s \\ n - s \end{matrix}. \quad (1)$$

The conditional distribution of  $\mathbf{Z}_1$  given  $\mathbf{Z}_2 = \mathbf{z}_2$  is also multivariate normal, given by

$$(\mathbf{Z}_1 \mid \mathbf{Z}_2 = \mathbf{z}_2) \sim N_s(\boldsymbol{\mu}_{1.2}, \mathbf{A}_{1.2}), \quad (2)$$

where

$$\boldsymbol{\mu}_{1.2} = \boldsymbol{\mu}_1 + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2),$$

and

$$\mathbf{A}_{1.2} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} .$$

Note that the conditional density of  $(\mathbf{Z}_1 | \mathbf{Z}_2 = \mathbf{z}_2)$  equals the ratio of the joint normal density of  $(\mathbf{Z}_1, \mathbf{Z}_2)$  and the marginal normal density of  $\mathbf{Z}_2$  evaluated at  $\mathbf{Z}_2 = \mathbf{z}_2$ . Taking  $\boldsymbol{\mu} = \mathbf{0}$  to simplify, cancelling constants, and taking the log of both sides, we get the useful result

$$(\mathbf{z}_1 - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{z}_2)^T(\mathbf{A}_{1.2})^{-1}(\mathbf{z}_1 - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{z}_2) = \mathbf{z}^T\mathbf{A}^{-1}\mathbf{z} - \mathbf{z}_2^T\mathbf{A}_{22}^{-1}\mathbf{z}_2 \quad (3)$$

which can be rewritten as

$$\mathbf{z}^T\mathbf{A}^{-1}\mathbf{z} = (\mathbf{z}_1 - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{z}_2)^T(\mathbf{A}_{1.2})^{-1}(\mathbf{z}_1 - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{z}_2) + \mathbf{z}_2^T\mathbf{A}_{22}^{-1}\mathbf{z}_2. \quad (4)$$

Amidst the cancellation of the constants is a useful relationship of the determinants:

$$|\mathbf{A}_{1.2}| = |\mathbf{A}| / |\mathbf{A}_{22}|. \quad (5)$$

We consider here the problem of computing the conditional mean vector  $\boldsymbol{\mu}_{1.2}$  and covariance matrix  $\mathbf{A}_{1.2}$  where  $n$  is large,  $s$  is modest in size, and where the partitioning is arbitrary. The applications envisioned include both generation from the conditional distribution and inference in cases of censored or missing values. Especially interesting are applications where the partitioning may be repeatedly changing, as may occur in some MCMC techniques. In Sections 2 and 3, we consider the case of the general multivariate normal distribution, and the special case where  $\mathbf{Z}$  arises from a Gaussian autoregressive-moving average (ARMA( $p, q$ )) process is considered in Section 4. FORTRAN 95 code implementing some of these results is included with this manuscript as ‘gsubex1.f95’ to ‘gsubex3.f95’.

## 2. A general approach with the sweep operator

Following standard computational techniques (see, for example, [Golub and van Loan 1996](#); [Monahan 2001](#); [Stewart 1973](#)) such as Gaussian elimination or Cholesky factorization, the direct computation of  $\boldsymbol{\mu}_{1.2}$  and  $\mathbf{A}_{1.2}$  will take  $O(n^3)$  floating point operations, owing from computing the inverse (solving a system of equations) in  $\mathbf{A}_{22}$  which is  $O(n)$  in size. However, if the partitioning were to be changed slightly, little of these results could be used to address the modified problem. The sweep operator is more appropriate here, being easy to update while taking nearly the same computations to form  $\boldsymbol{\mu}_{1.2}$  and  $\mathbf{A}_{1.2}$ .

The sweep operator is widely used in regression problems (see [Stiefel 1963](#); [Goodnight 1979](#); [Monahan 2001](#), Section 5.12) computing the following tableau after sweeping the first  $p$  rows/columns of the  $(p+1) \times (p+1)$  matrix:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{y} \\ \mathbf{y}^T\mathbf{X} & \mathbf{y}^T\mathbf{y} \end{bmatrix} \rightarrow \text{sweep} \rightarrow \begin{bmatrix} (\mathbf{X}^T\mathbf{X})^{-1} & (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ -\mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} & \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \end{bmatrix} .$$

The sweep operator can be viewed as a compact scheme for full column elimination using diagonal pivots  $v_{jj}$ . It is usually implemented as a single step, operating on one row/column at a time, so that the inverse is a reciprocal. When an identity matrix is augmented, as below, full elimination pivoting on the first  $p$  columns produces the matrix

$$\begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{y} & \mathbf{I}_p \\ \mathbf{y}^T\mathbf{X} & \mathbf{y}^T\mathbf{y} & \mathbf{0} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{I}_p & (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} & (\mathbf{X}^T\mathbf{X})^{-1} \\ \mathbf{0} & \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} & -\mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \end{bmatrix} .$$

Merely overwriting the inverse on top of the matrix, and avoiding storing the known elements of the identity matrix produces the sweep step. The sweep operator can be used to compute the regression coefficients  $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$  and error sum of squares  $\mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ . Applying the sweep operator to the partitioned covariance matrix  $\mathbf{A}$ , the result of sweeping the LAST  $n - s$  rows/columns produced the following tableau

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \rightarrow \text{sweep} \rightarrow \begin{bmatrix} \mathbf{A}_{1,2} & -\mathbf{A}_{12} \mathbf{A}_{22}^{-1} \\ \mathbf{A}_{22}^{-1} \mathbf{A}_{21} & \mathbf{A}_{22}^{-1} \end{bmatrix}$$

which obviously includes the necessary information for the conditional distribution. The ease of computing updates follows from the commutativity and reversibility properties of the sweep operator. The commutativity property means that the order of the sweeping does not matter. Reversibility means that sweeping a particular row/column a second time is the same as not sweeping it in the first place. As a consequence, sweeping a particular row/column merely moves a variable from one subset to the other. For the conditional distribution, the rows/columns corresponding to the variables that are to be conditioned upon are swept. Conditioning on a new variable means sweeping that row/column of the covariance matrix. Removing a variable from the list to be conditioned upon means sweeping that row/column a second time. Each sweep of a row/column takes  $O(n^2)$  operations.

As in computing the determinant from full column pivoting, the determinant of  $\mathbf{A}_{22}$  can be computed from the product of the pivot elements  $v_{jj}$ , which can be seen in the following Example 1.

Example 1: Let  $n = 3$ , and  $\mathbf{A}$  given below and the following sweep sequence:

$$\begin{aligned} \mathbf{A} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 6 \end{bmatrix} &\rightarrow \text{sweep } 2 \rightarrow \begin{bmatrix} 2.75 & -.25 & -.5 \\ .25 & .25 & .5 \\ -.5 & -.5 & 5 \end{bmatrix} \quad v_{22} = 4, \quad \left| \begin{bmatrix} 4 \end{bmatrix} \right| = 4 \\ (Z_1, Z_3 \mid Z_2 = z_2) &\sim N_2 \left( \begin{bmatrix} \mu_1 \\ \mu_3 \end{bmatrix} + \begin{bmatrix} 1/4 \\ 1/2 \end{bmatrix} (z_2 - \mu_2), \begin{bmatrix} 2.75 & -.5 \\ -.5 & 5 \end{bmatrix} \right) \\ \begin{bmatrix} 2.75 & -.25 & -.5 \\ .25 & .25 & .5 \\ -.5 & -.5 & 5 \end{bmatrix} &\rightarrow \text{sweep } 3 \rightarrow \begin{bmatrix} 2.7 & -.3 & .1 \\ .3 & .3 & -.1 \\ -.1 & -.1 & .2 \end{bmatrix} \quad v_{33} = 5, \quad \left| \begin{bmatrix} 4 & 2 \\ 2 & 6 \end{bmatrix} \right| = 20 \\ (Z_1 \mid Z_2 = z_2, Z_3 = z_3) &\sim N_1(\mu_1 + \begin{bmatrix} .3 & -.1 \end{bmatrix} \begin{bmatrix} z_2 - \mu_2 \\ z_3 - \mu_3 \end{bmatrix}, 2.7) \\ \begin{bmatrix} 2.7 & -.3 & -.1 \\ .3 & .3 & -.1 \\ -.1 & -.1 & .2 \end{bmatrix} &\rightarrow \text{sweep } 2 \rightarrow \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3.333 & -.333 \\ 0 & .333 & .166 \end{bmatrix} \quad v_{22} = .3, \quad \left| \begin{bmatrix} 6 \end{bmatrix} \right| = 6 \\ (Z_1, Z_2 \mid Z_3 = z_3) &\sim N_2 \left( \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/3 \end{bmatrix} (z_3 - \mu_3), \begin{bmatrix} 3 & 1 \\ 1 & 3.333 \end{bmatrix} \right) \\ \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3.333 & -.333 \\ 0 & .333 & .166 \end{bmatrix} &\rightarrow \text{sweep } 1 \rightarrow \begin{bmatrix} .333 & .333 & 0 \\ -.333 & 3 & -.333 \\ 0 & .333 & .166 \end{bmatrix} \quad v_{11} = 3, \quad \left| \begin{bmatrix} 3 & 0 \\ 0 & 6 \end{bmatrix} \right| = 18 \\ (Z_2 \mid Z_1 = z_1, Z_3 = z_3) &\sim N_1(\mu_2 + \begin{bmatrix} 1/3 & 1/3 \end{bmatrix} \begin{bmatrix} z_1 - \mu_1 \\ z_3 - \mu_3 \end{bmatrix}, 3) \end{aligned}$$

Note that the repeating decimals .333 and .166 in the example above are rounded versions of  $1/3$  and  $1/6$ . In practice, rounding error will accumulate with each sweep step, and the process may need to be restarted when the accumulated error becomes noticeable.

### 3. A second general method

This second general approach aims to solve the two specific problems without explicitly computing the conditional mean vector and covariance matrix. One task is computing the likelihood of the conditional normal distribution, namely the quadratic form

$$(\mathbf{z}_1 - \boldsymbol{\mu}_{1,2})^T (\mathbf{A}_{1,2})^{-1} (\mathbf{z}_1 - \boldsymbol{\mu}_{1,2})$$

and the determinant  $|\mathbf{A}_{1,2}|$ . The second task would be to generate a random vector from the conditional distribution.

Consider the reverse-Cholesky factorization of the covariance matrix  $\mathbf{A}$ , producing the upper triangular factor  $\mathbf{U}$ :

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \mathbf{U}\mathbf{U}^T = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{bmatrix}^T = \begin{bmatrix} \mathbf{U}_1\mathbf{U}_1^T + \mathbf{U}_2\mathbf{U}_2^T & \mathbf{U}_2\mathbf{U}_3^T \\ \mathbf{U}_3\mathbf{U}_2^T & \mathbf{U}_3\mathbf{U}_3^T \end{bmatrix}$$

The usual Cholesky factorization produces a lower triangular matrix and the induction order goes from upper left to lower right; here the order is reversed and an upper triangular matrix is produced. Now consider the problem of generating from the joint distribution using a random vector

$$\mathbf{v} \sim \text{Normal}_n(\mathbf{0}, \mathbf{I}_n).$$

Partitioning  $\mathbf{v}$  in the same way, the algebra would follow the simple form

$$\begin{aligned} \mathbf{z}_1 &= \boldsymbol{\mu}_1 + \mathbf{U}_1\mathbf{v}_1 + \mathbf{U}_2\mathbf{v}_2 \\ \mathbf{z}_2 &= \boldsymbol{\mu}_2 + \mathbf{U}_3\mathbf{v}_2. \end{aligned}$$

Conditioning on  $\mathbf{Z}_2 = \mathbf{z}_2$  would really mean knowing  $\mathbf{z}_2$ . If that were so, we could then solve for  $\mathbf{v}_2$ , solving the triangular system to get

$$\mathbf{v}_2 = \mathbf{U}_3^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2).$$

Then  $\mathbf{z}_1$  could then be generated conditional on the value of  $\mathbf{z}_2$ , following

$$\mathbf{z}_1 = \boldsymbol{\mu}_1 + \mathbf{U}_1\mathbf{v}_1 + \mathbf{U}_2\mathbf{U}_3^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2),$$

leading to the result  $\mathbf{A}_{1,2} = \mathbf{U}_1\mathbf{U}_1^T$ .

For computing the likelihood, the determinant part is easy:  $|\mathbf{A}_{1,2}| = |\mathbf{U}_1|^2$ , following from (5), since  $|\mathbf{U}_3|^2 = |\mathbf{A}_{22}|$ . The quadratic form is nearly as easy, since from (4) we have

$$(\mathbf{z}_1 - \boldsymbol{\mu}_{1,2})^T (\mathbf{A}_{1,2})^{-1} (\mathbf{z}_1 - \boldsymbol{\mu}_{1,2}) = (\mathbf{z} - \boldsymbol{\mu})^T \mathbf{A}^{-1} (\mathbf{z} - \boldsymbol{\mu}) - (\mathbf{z}_2 - \boldsymbol{\mu}_2)^T (\mathbf{A}_{22})^{-1} (\mathbf{z}_2 - \boldsymbol{\mu}_2).$$

The quadratic form in  $\mathbf{A}^{-1}$  can be written in partitioned form as the squared length of

$$\begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z}_1 - \boldsymbol{\mu}_1 \\ \mathbf{z}_2 - \boldsymbol{\mu}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1^{-1}[(\mathbf{z}_1 - \boldsymbol{\mu}_1) - \mathbf{U}_2\mathbf{U}_3^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2)] \\ \mathbf{U}_3^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2) \end{bmatrix},$$

while the quadratic form in  $(\mathbf{A}_{22})^{-1}$  is just the squared length of the second partitioned vector above

$$(\mathbf{z}_2 - \boldsymbol{\mu}_2)^T (\mathbf{A}_{22})^{-1} (\mathbf{z}_2 - \boldsymbol{\mu}_2) = \left\| \mathbf{U}_3^{-1} (\mathbf{z}_2 - \boldsymbol{\mu}_2) \right\|^2$$

so that the difference is the squared length of the first component

$$(\mathbf{z}_1 - \boldsymbol{\mu}_{1.2})^T (\mathbf{A}_{1.2})^{-1} (\mathbf{z}_1 - \boldsymbol{\mu}_{1.2}) = \left\| \mathbf{U}_1^{-1} [(\mathbf{z}_1 - \boldsymbol{\mu}_1) - \mathbf{U}_2 \mathbf{U}_3^{-1} (\mathbf{z}_2 - \boldsymbol{\mu}_2)] \right\|^2.$$

As presented, this approach offers no particular advantage over any other direct method – it merely uses a form of Cholesky decomposition. If the partitioning were changed, however, this Cholesky factor can be updated. The updating procedure due to [Clarke \(1981\)](#) was devised for subset regression computations and follows the same spirit as the sweep operator (see also [Daniel, Gragg, Kaufman, and Stewart 1976](#)). The efficiency of the update scheme will depend on the pattern of changes of partitioning.

The change in partitioning can be viewed in terms of a permutation matrix  $\mathbf{P}$ . Conditioning on an additional component (or one fewer) means permuting that component to the partition boundary and then moving the boundary. Here we want to construct the reverse-Cholesky (upper triangular) factor of the changed matrix  $\mathbf{PAP}^T$  using  $\mathbf{PU}$ . The problem is that  $\mathbf{PU}$  is no longer upper triangular but needs to be returned to that form. This can be done by postmultiplying by sufficient Givens transformations  $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_m$ , each an orthogonal matrix, as many as may be needed to return  $\mathbf{PUG}_1 \cdots \mathbf{G}_m$  to upper triangular form, and restoring the factorization

$$\mathbf{PAP}^T = (\mathbf{PUG}_1 \cdots \mathbf{G}_m) (\mathbf{PUG}_1 \cdots \mathbf{G}_m)^T.$$

Example 2: For the covariance matrix  $\mathbf{A}$  below, the decomposition

$$\mathbf{A} = \begin{bmatrix} 25 & 11 & 0 & 4 \\ 11 & 19 & 7 & 2 \\ 0 & 7 & 5 & 2 \\ 4 & 2 & 2 & 4 \end{bmatrix} = \mathbf{UU}^T = \begin{bmatrix} 2 & 4 & -1 & 2 \\ 0 & 3 & 3 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 4 & 3 & 0 & 0 \\ -1 & 3 & 2 & 0 \\ 2 & 1 & 1 & 2 \end{bmatrix}$$

provides  $\mathbf{U}$  that could be easily employed for  $(Z_1, Z_2 | Z_3 = z_3, Z_4 = z_4)$ . Now suppose we now want  $(Z_1, Z_4 | Z_2 = z_2, Z_3 = z_3)$ , so that  $\mathbf{A}_{1.2} = \begin{bmatrix} 11.8478 & 4.9565 \\ 4.9565 & 3.1304 \end{bmatrix}$ ,  $|\mathbf{A}|_{1.2} = |\mathbf{A}|/|\mathbf{A}_{22}| = 596/46 = 12.5217$ , and

$$\mathbf{PU} = \begin{bmatrix} 2 & 4 & -1 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 3 & 3 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix}.$$

To return this matrix to upper triangular form, first postmultiply by a Givens transformation  $\mathbf{G}_1$  defined by elements (4,3) and (4,4), changing everything in columns 3 and 4, and placing a zero in (4,3). Here  $\mathbf{G}_1$  is an identity matrix except for  $(\mathbf{G}_1)_{33} = (\mathbf{G}_1)_{44} = 1/\sqrt{5}$  and  $(\mathbf{G}_1)_{34} = -(\mathbf{G}_1)_{43} = 2/\sqrt{5}$ . Next construct  $\mathbf{G}_2$  using (3,2) and (3,3), changing columns 2 and 3 and placing a zero in (3,2), and return the upper triangular form. The zeros in (2,2) and (2,3) will change with these two transformations, but this will not affect the triangular form.

#### 4. ARMA covariance model

The case where  $Z_t, t = 1, \dots, n$  follows the ARMA( $p, q$ ) time series model

$$Z_t - \phi_1 Z_{t-1} \dots - \phi_p Z_{t-p} = e_t - \theta_1 e_{t-1} \dots - \theta_q e_{t-q} \quad (6)$$

brings some special structure that can be exploited. If the partitioning followed the same simple structure as in (1), then the conditional distribution can be easily constructed. However, we are interested here in the case where the partitioned vectors are

$$\mathbf{Z}_1 = (Z_{S(1)}, Z_{S(2)}, \dots, Z_{S(s)})^T \text{ and } \mathbf{Z}_2 = (Z_{T(1)}, Z_{T(2)}, \dots, Z_{T(n-s)})^T$$

and where subset list of indices  $S(j)$  may be arbitrary and  $S$  and  $T$  are complementary.

A device due to Ansley can be exploited to compute the conditional distribution efficiently. Let  $Z_t, t = 1, \dots, n$  follow the ARMA( $p, q$ ) process (6) and denote the covariance matrix by  $\mathbf{A}$ . Now let the  $n \times n$  matrix  $\mathbf{B}$  have its first  $p$  rows all zeros except for 1 on the diagonal, and the last  $n - p$  rows of the form

$$\dots \quad 0 \quad 0 \quad -\phi_p \quad -\phi_{p-1} \dots \quad -\phi_2 \quad -\phi_1 \quad 1 \quad 0 \quad 0 \quad \dots$$

aligned so that the 1 appears on the diagonal. Ansley (1979) shows that the covariance matrix of  $\mathbf{BZ}$ , namely  $\mathbf{BAB}^T$ , is a banded positive definite matrix. The bandwidth of  $\mathbf{BAB}^T$  is  $\max(p, q) = m$  for the first  $p$  rows, and  $q$  thereafter. Its Cholesky factor  $\mathbf{L}$ , such that

$$\mathbf{BAB}^T = \mathbf{LL}^T,$$

is lower triangular with the same banding structure, and can be computed in  $O(nm^2)$ . Solving a system of equations in  $\mathbf{L}$ , say to compute  $\mathbf{L}^{-1}\mathbf{u}$ , can be done in  $O(nm)$ . The space required is only  $O(m^2)$ . As a result, a bilinear form in the inverse of the covariance matrix

$$\mathbf{u}^T \mathbf{A}^{-1} \mathbf{v} = \mathbf{u}^T (\mathbf{B}^T \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{B}) \mathbf{v} = (\mathbf{L}^{-1} \mathbf{B} \mathbf{u})^T (\mathbf{L}^{-1} \mathbf{B} \mathbf{v})$$

can be computed in  $O(nm^2)$  time and  $O(m^2)$  space.

Before applying this result to the arbitrary partitioning problem, and so following the original partitioning in (1), first notice that

$$\begin{bmatrix} \mathbf{I}_s \\ \mathbf{0} \end{bmatrix} \mathbf{A}^{-1} \begin{bmatrix} \mathbf{I}_s & \mathbf{0} \end{bmatrix} = (\mathbf{A}_{1.2})^{-1} \quad (7)$$

which can be established through the usual results for the inverse of a partitioned matrix. The remaining results are more complicated. Defining  $Q(\mathbf{z}) = \mathbf{z}^T \mathbf{A}^{-1} \mathbf{z}$ , and employing result (4), we have the derivative results

$$\partial^2 Q(\mathbf{z}) / \partial z_i \partial z_j = 2 \left[ (\mathbf{A}_{1.2})^{-1} \right]_{ij}$$

for  $1 \leq i, j \leq s$ , and

$$\partial Q(\mathbf{z}) / \partial z_i |_{z_1=0} = -2 \left[ (\mathbf{A}_{1.2})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{z}_2 \right]_i$$

for  $1 \leq i \leq s$ . For computation, we replace the derivative with a difference expression. In the case of a quadratic function  $Q(\mathbf{z})$ , the second difference will be exact for the second partial derivative, so that

$$[Q(\mathbf{z} + \delta \mathbf{e}_i + \delta \mathbf{e}_j) - Q(\mathbf{z} + \delta \mathbf{e}_i) - Q(\mathbf{z} + \delta \mathbf{e}_j) + Q(\mathbf{z})] / \delta^2 = \partial^2 Q(\mathbf{z}) / \partial z_i \partial z_j = 2 \mathbf{e}_i^T \mathbf{A}^{-1} \mathbf{e}_j$$

or

$$\mathbf{e}_i^T \mathbf{A}^{-1} \mathbf{e}_j = [(\mathbf{A}_{1.2})^{-1}]_{ij}$$

where  $\mathbf{e}_i$  is the  $i^{\text{th}}$  elementary vector, verifying (7) above. A similar result using the first difference, at  $\mathbf{z}_1 = \mathbf{0}$ , gives

$$\begin{aligned} [Q\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix} + \delta \mathbf{e}_i\right) - Q\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix}\right)] / \delta &= \partial Q(\mathbf{z}) / \partial z_i |_{z_1=0} \\ &= -2 [(\mathbf{A}_{1.2})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{z}_2]_i \end{aligned}$$

leading to the computational route of

$$\mathbf{e}_i^T \mathbf{A}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix} = [(\mathbf{A}_{1.2})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{z}_2]_i$$

for  $1 \leq i \leq s$ .

Now to transform these results to the case of arbitrary partitioning, construct the  $n \times s$  matrix  $\mathbf{P}$ , which is zero everywhere except for elements

$$(\mathbf{P})_{S(i),i} = 1$$

for  $1 \leq i \leq s$ , and, similarly, the  $n \times (n - s)$  matrix  $\mathbf{Q}$ , which is zero everywhere except

$$(\mathbf{Q})_{T(i),i} = 1$$

for  $1 \leq i \leq n - s$ . Putting  $\mathbf{P}$  and  $\mathbf{Q}$  together as  $\begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix}$  forms an  $n \times n$  permutation matrix. Then we have the computational formulas

$$\mathbf{P}^T \mathbf{A}^{-1} \mathbf{P} = (\mathbf{L}^{-1} \mathbf{B} \mathbf{P})^T (\mathbf{L}^{-1} \mathbf{B} \mathbf{P}) = (\mathbf{A}_{1.2})^{-1}$$

and

$$\begin{aligned} \mathbf{P}^T \mathbf{A}^{-1} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix} &= (\mathbf{L}^{-1} \mathbf{B} \mathbf{P})^T (\mathbf{L}^{-1} \mathbf{B} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix}) \\ &= -(\mathbf{A}_{1.2})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{z}_2. \end{aligned}$$

The key computational advantage of this method is that, as bilinear forms in  $\mathbf{A}^{-1}$ , the computational burden is only  $O(nm^2 + nms + s^2)$  for ARMA( $p, q$ ) time series models.

One remaining detail is that  $(\mathbf{A}_{1.2})^{-1}$  and  $(\mathbf{A}_{1.2})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{z}_2$  (or with  $(\mathbf{z}_2 - \boldsymbol{\mu}_2)$ ) are not the most convenient forms for most applications, which would involve  $\mathbf{A}_{1.2}$  or  $(\mathbf{A}_{1.2})^{-1}$ , and

$\mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2)$ . Equally useful would be a Cholesky factor of  $(\mathbf{A}_{1.2})^{-1}$  and  $\mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2)$ . To compute these in a stable and efficient manner, construct the matrix

$$\begin{bmatrix} L^{-1}\mathbf{B}\mathbf{P} & L^{-1}\mathbf{B}\mathbf{P} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \\ \mathbf{0} \\ \mathbf{z}_2 - \boldsymbol{\mu}_2 \end{bmatrix} = L^{-1}\mathbf{B} \begin{bmatrix} \mathbf{P} & \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \\ \mathbf{0} \\ \mathbf{z}_2 - \boldsymbol{\mu}_2 \end{bmatrix}$$

and perform Householder transformations  $\mathbf{H}_1, \dots, \mathbf{H}_s$

$$\mathbf{H}_s \cdots \mathbf{H}_2 \mathbf{H}_1 \begin{bmatrix} L^{-1}\mathbf{B}\mathbf{P} & L^{-1}\mathbf{B} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \\ \mathbf{0} \\ \mathbf{z}_2 - \boldsymbol{\mu}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{u} \end{bmatrix}$$

to form the upper triangular matrix  $\mathbf{R}$ . Then  $\mathbf{R}^T\mathbf{R} = (\mathbf{A}_{1.2})^{-1}$  shows that  $\mathbf{R}$  is a factor, and  $\mathbf{R}^{-1}\mathbf{u} = \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2)$  finishes the computations. Computation of the likelihood from the conditional distribution would then take the computational route of

$$(\mathbf{z}_1 - \boldsymbol{\mu}_{1.2})^T (\mathbf{A}_{1.2})^{-1} (\mathbf{z}_1 - \boldsymbol{\mu}_{1.2}) = [\mathbf{R}(\mathbf{z}_1 - \boldsymbol{\mu}_{1.2})]^T [\mathbf{R}(\mathbf{z}_1 - \boldsymbol{\mu}_{1.2})]$$

and generation from the conditional distribution would follow

$$\mathbf{z}_1 = \boldsymbol{\mu}_{1.2} + \mathbf{R}^{-1}\mathbf{v},$$

where, as before,  $\mathbf{v} \sim \text{Normal}_n(\mathbf{0}, \mathbf{I}_n)$ . The steps using Householder transformations are the same orthogonalization steps for a QR factorization in regression; see, for example, [Golub and van Loan \(1996, Section 6.2\)](#), [Monahan \(2001, Section 5.6\)](#), or [Stewart \(1973, Section 5.3\)](#).

Finally, if the conditioning variables  $\mathbf{z}_2$  (or its mean) were changed, the resulting vector  $\mathbf{u}$  would have to be recomputed, taking only  $O(n)$ . If a variable were moved from  $S$  to  $T$ , the mean vector would change and  $\mathbf{u}$  must be updated. If a variable were moved from  $T$  to  $S$ , then a new column would be added to  $\mathbf{P}$ , and the mean vector would also have to be updated.

## Acknowledgements

The author thanks the referee for careful reading and checking the code and Sujit Ghosh for suggesting the problem.

## References

- Ansley CF (1979). "An Algorithm for the Exact Likelihood of a Mixed Autoregressive Moving Average Process." *Biometrika*, **66**, 59–65.
- Clarke MRB (1981). "AS 163: A Givens Algorithm for Moving from One Linear Model to Another Without Going Back to the Data." *Applied Statistics*, **30**, 198–203.
- Daniel JW, Gragg WB, Kaufman L, Stewart GW (1976). "Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization." *Mathematics of Computation*, **30**, 772–795.
- Golub GH, van Loan C (1996). *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3rd edition.



- Goodnight JH (1979). “A Tutorial on the Sweep Operator.” *The American Statistician*, **33**, 149–150.
- Monahan JF (2001). *Numerical Methods of Statistics*. Cambridge University Press, New York.
- Stewart GW (1973). *Introduction to Matrix Computations*. Academic Press, New York.
- Stiefel EL (1963). *An Introduction to Numerical Mathematics*. Academic Press, New York.

**Affiliation:**

John F. Monahan  
Department of Statistics  
North Carolina State University  
Raleigh, NC 27695-8203, United States of America  
E-mail: [monahan@stat.ncsu.edu](mailto:monahan@stat.ncsu.edu)  
URL: <http://www.stat.ncsu.edu/~monahan/>