MATHEMATICS OF COMPUTATION
VOLUME 40, NUMBER 161
JANUARY 1983, PAGES 323–336

A General Orthogonalization Technique With Applications to Time Series Analysis and Signal Processing*

By George Cybenko

Abstract. A new orthogonalization technique is presented for computing the QR factorization of a general $n \times p$ matrix of full rank p $(n \geq p)$. The method is based on the use of projections to solve increasingly larger subproblems recursively and has an $O(np^2)$ operation count for general matrices. The technique is readily adaptable to solving linear least-squares problems. If the initial matrix has a circulant structure the algorithm simplifies significantly and gives the so-called lattice algorithm for solving linear prediction problems. From this point of view it is seen that the lattice algorithm is really an efficient way of solving specially structured least-squares problems by orthogonalization as opposed to solving the normal equations by fast Toeplitz algorithms.

1. Introduction. The QR decomposition (also called the orthogonal decomposition, factorization, and triangularization) of an $n \times p$ matrix, X $(n \geq p)$

$$X = QR, \tag{1.1}$$

where Q is $n \times p$ with orthonormal columns and R is $p \times p$ and upper triangular has proven to be useful in solving a number of linear algebraic problems [2], [10], [18], [19]. The standard algorithms for computing this QR decomposition include reductions by orthogonal transformations (Householder and Givens methods) and Gram-Schmidt orthogonalization (including the modified Gram-Schmidt method) [18]. All these methods use $O(np^2)$ arithmetic operations.

In this paper, a new method of computing the QR decomposition is described. This algorithm in fact explicitly computes a Q with only orthogonal columns and the corresponding R^{-1}, which represents only minor differences with the standard algorithms since Q can be easily normalized while R^{-1} is of interest in many applications, and R is computable from R^{-1} in $O(p^3)$ operations. The method is based on the recursive solution by projection to least-squares problems involving contiguous blocks of columns of X, and as such, bears resemblence to the construction of forward and backward innovations processes as described by Kailath in the context of time series and signal processing [13]. This method is also closely related to a novel method of matrix inverse triangular decomposition [7] in that the method described in that work is the normal equations analogue of our algorithm for orthogonalization.

Received December 30, 1980; revised March 30, 1982.
1980 *Mathematics Subject Classification.* Primary 65F25; Secondary 62M20.
Key words and phrases. Orthogonalization, least-squares problems, linear prediction.
*This work was supported in part by the National Science Foundation under Grants MCS-7817697 and MCS-8003364.

0025-5718/82/0000-0749/$04.75

Time and space complexities of this method for general matrices, X, are of the same order of magnitude as the above-mentioned standard algorithms, but with larger constants. In light of this fact, the algorithm presented here is not recommended as a means of computing QR decompositions of general matrices with conventional computing machinery. However, it will be seen that if the matrix X has a Toeplitz structure with "zero boundaries" (to be explained in the following) or more generally, is circulant, the algorithm reduces to an extremely efficient $O(np)$ method taking full advantage of the circulant structure. In the special Toeplitz case, it will be shown that the streamlined algorithm is precisely the so-called Itakura-Saito-Burg lattice method recently popularized in the study of linear predictive coding of speech and deconvolution of seismic data [4], [12], [14]. That algorithm was independently derived by De Meersman in [8]. Another possible advantage is the degree of parallelism to which the algorithm seems amenable.

In a sense it is unfair to measure the complexity of this method with other algorithms computing the QR decomposition, since this algorithm explicitly computes the orthogonal decompositions of all subblocks of contiguous columns of the matrix X and as such gives much more. Potential application of this aspect towards the problem of subset regression [9], [1] demands attention, although no significant results are available now, save the observation that [9] uses a branch-and-bound method on the partial order of all regression subsets so that a significant amount of pruning may be done early given the large number of computed subset least-squares solutions obtained by our algorithm.

Section 2 contains a description of the algorithm for a general matrix together with a time and space complexity analysis. In that section, a method for solving least-squares problems is presented also. In Section 3, the algorithm is refined to deal with special matrix structures, where columns are unitarily related. It will be seen there that the algorithm simplifies significantly, but when the unitary operator is given by the circular shift, the economization is even more dramatic. The case of the circular shift operator, which gives rise to the circulant structure alluded to above, is taken up in Section 4 which describes the application to the linear prediction of stationary time series. That section contains a brief comparison with respect to conditioning and complexity between this method and the autocorrelation method for solving linear prediction and Wiener filtering problems. Section 5 is a summary.

2. An Algorithm for Computing QR Decompositions. Let X be an $n \times p$ matrix of full rank p $(n \geq p)$ with columns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$. The matrix consisting of the $j - i + 1$ columns $(i \leq j)$ $\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_j$ in that order is denoted by $X(i, j)$. The span of the columns of a matrix M is denoted by $L(M)$.

From the theory of least-squares approximation, it is well known that the following two problems have identical solutions:

–Determine a $j - i + 1$ vector $\mathbf{a}$ so that

$$X(i, j)\mathbf{a} + \mathbf{x}_{j+1} \tag{2.1}$$

is orthogonal to $L(X(i, j))$.

–Determine a $j - i + 1$ vector $\mathbf{a}$ so that the 2-norm of

$$X(i, j)\mathbf{a} + \mathbf{x}_{j+1} \tag{2.2}$$

is minimal over all such vectors $\mathbf{a}$.

Since the vector $\mathbf{a}$ simultaneously solves both problems uniquely, both characterizations will be used without further explanation in the following.

Now define $\mathbf{f}_{k+1}$ as the vector in (2.1) and hence (2.2), with $\mathbf{f}_1 = \mathbf{x}_1$. We then have that

$$\mathbf{f}_k \in L(X(1, k)), \qquad \mathbf{f}_k \perp L(X(1, k-1)),$$

and so $\mathbf{f}_i \perp \mathbf{f}_j$ providing $i \neq j$.

Thus the matrix

$$Q = (\mathbf{f}_1, \ldots, \mathbf{f}_p)$$

has orthogonal, but not necessarily orthonormal, columns.

Furthermore defining the upper triangular matrix T as

$$T = \begin{bmatrix} 1 & a_{1,1} & a_{2,2} & \cdot & \cdot & \cdot & \cdot & a_{p-1,p-1} \\ 0 & 1 & a_{2,1} & \cdot & \cdot & \cdot & \cdot & a_{p-1,p-2} \\ 0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & \ddots & & & & \cdot \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & a_{p-1,1} \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & & 1 \end{bmatrix}$$

where $\mathbf{a} = (a_{k,k}, a_{k,k-1}, \ldots, a_{k,1})^T$ is the solution to (2.1) and (2.2) with $i = 1$ and $j = k$, it is evident that

$$XT = Q \tag{2.3}$$

or, equivalently,

$$X = QT^{-1}. \tag{2.4}$$

This last equation shows that Q and T^{-1} yield a QR decomposition of X once the columns of Q are normalized and T^{-1} is scaled appropriately. This discussion has shown that the QR decomposition is intimately associated with least-squares problems involving contiguous blocks of columns of X. The basis of the method described here rests on the ability to piece together solutions to such least-squares problems to form solutions to larger least-squares problems.

In particular, it will now be shown that by considering a larger class of least-squares problems, a class containing (2.2), the entries of T and the residuals $\mathbf{f}_k$ are easily computable. In order to delay the introduction of indices and subscripts, we shall present the basic idea in the form of a general theorem.

THEOREM. *Suppose that A is an $n \times k$ real matrix and that* $\mathbf{b}$ *and* $\mathbf{c}$ *are n-vectors so that the augmented matrix* $(\mathbf{b}, A, \mathbf{c})$ *has full rank $k + 2$. Let* $\mathbf{x}$ *and* $\mathbf{w}$ *be k-vectors such that the n-vectors* $\mathbf{r}$ *and* $\mathbf{s}$ *given by*

$$\mathbf{r} = A\mathbf{x} + \mathbf{b} \quad \textit{and} \quad \mathbf{s} = A\mathbf{w} + \mathbf{c}$$

are the residuals in solutions to the least-squares problems

$$\min\|A\mathbf{x} + \mathbf{b}\|_2, \qquad \min\|A\mathbf{w} + \mathbf{c}\|_2,$$

respectively. (In case A is vacuous, we let $\mathbf{r} = \mathbf{b}$ *and* $\mathbf{s} = \mathbf{c}$.) *Defining*

$$\alpha = -\frac{\mathbf{r}^T\mathbf{s}}{\mathbf{r}^T\mathbf{r}}, \qquad \beta = -\frac{\mathbf{r}^T\mathbf{s}}{\mathbf{s}^T\mathbf{s}},$$

we have that

$$\mathbf{z} = \begin{bmatrix} \alpha \\ \mathbf{w} + \alpha\mathbf{x} \end{bmatrix}$$

solves the least-squares problem

$$\min\|(\mathbf{b}, A)\mathbf{z} + \mathbf{c}\|_2, \tag{2.5}$$

while

$$\mathbf{v} = \begin{bmatrix} \mathbf{x} + \beta\mathbf{w} \\ \beta \end{bmatrix}$$

solves the least-squares problem

$$\min\|(A, \mathbf{c})\mathbf{v} + \mathbf{b}\|_2.$$

The residuals for the above two least-squares solutions are

$$\mathbf{s} + \alpha\mathbf{r} \quad \textit{and} \quad \mathbf{r} + \beta\mathbf{s},$$

respectively.

Proof. Notice that $(\mathbf{b}, A)\mathbf{z} + \mathbf{c} = \alpha\mathbf{r} + \mathbf{s}$. Since $\mathbf{r}$ and $\mathbf{s}$ are residuals in least-squares solutions involving A, both $\mathbf{r}$ and $\mathbf{s}$ are orthogonal to $L(A)$ and so the residual $\alpha\mathbf{r} + \mathbf{s}$ is also orthogonal to $L(A)$. Thus in order that $\alpha\mathbf{r} + \mathbf{s}$ be the residual in the least-squares solution to (2.5), it is only necesssary to show that $\alpha\mathbf{r} + \mathbf{s}$ is orthogonal to $\mathbf{b}$.

This is the case, providing

$$\alpha = -\frac{\mathbf{b}^T\mathbf{s}}{\mathbf{b}^T\mathbf{r}}.$$

Noting the definition of $\mathbf{r}$ and the fact that $\mathbf{r}$ is orthogonal to $L(A)$ shows that

$$\mathbf{b}^T\mathbf{r} = \mathbf{r}^T\mathbf{r} \quad \text{and} \quad \mathbf{b}^T\mathbf{s} = \mathbf{r}^T\mathbf{s},$$

thereby establishing the result for the augmented matrix $(\mathbf{b}, A)$. The same argument gives the result for the augmented matrix $(A, \mathbf{c})$. □

It is straightforward to see now how the method of the above theorem fits into our computation of the factorization $XT = Q$. As observed before, T and Q are obtained from least-squares solutions to problems of the type

$$\min\|X(1, k)\mathbf{v} + \mathbf{x}_{k+1}\|_2.$$

The theorem tells us how to use solutions to

$$\min\|X(i, j)\mathbf{u} + \mathbf{x}_{j+1}\|_2 \quad \text{and} \quad \min\|X(i, j)\mathbf{v} + \mathbf{x}_{i-1}\|_2$$

to compute solutions to

$$\min\|X(i-1, j)\mathbf{w} + \mathbf{x}_{j+1}\|_2 \quad \text{and} \quad \min\|X(i, j+1)\mathbf{y} + \mathbf{x}_{i-1}j\|_2.$$

The algorithm essentially consists of sweeping across the columns of X solving the least-squares problems involving single columns of X (this is the initialization). These solutions are then combined according to the theorem to give solutions to the least-squares problems involving pairs of contiguous columns of X, and these in turn are used to construct solutions to problems involving triples of contiguous columns of X and so on.

In order to make the algorithm's formal description more meaningful, we shall describe the roles of the various quantities in it first.

For each pair of integers (i, j) with $1 \leqslant i \leqslant j \leqslant p$ there are two least-squares problems determined by the columns of X.

–The (i, j) backward approximation. This is the least-squares problem of approximating $\mathbf{x}_i$ by a vector from $L(X(i+1, j))$. Specifically, we wish to find coefficients $\beta_m^{(i,j)}$ so that the residual

$$\mathbf{b}^{(i,j)} = \mathbf{x}_i + \beta_1^{(i,j)}\mathbf{x}_{i+1} + \cdots + \beta_{j-1}^{(i,j)}\mathbf{x}_j$$

has minimal 2-norm (or equivalently, is orthogonal to $L(X(i+1, j))$).

–The (i, j) forward approximation. This is the least-squares problem of approximating $\mathbf{x}_j$ by a vector from $L(X(i, j-1))$. Specifically, we wish to find coefficients $\phi_m^{(i,j)}$ so that the residual

$$\mathbf{f}(i, j) = \phi_{j-i}^{(i,j)}\mathbf{x}_i + \cdots + \phi_1^{(i,j)}\mathbf{x}_{j-1} + \mathbf{x}_j$$

has minimal 2-norm (or equivalently, is orthogonal to $L(X(i, j-1))$).

The algorithm described below is based on the properties and observations detailed above. The computation of the coefficients $\beta_m^{(i,j)}$, $\phi_m^{(i,j)}$ and residuals $\mathbf{f}^{(i,j)}$, $\mathbf{b}^{(i,j)}$ is then summarized as follows:

Initialization: $\mathbf{f}^{(i,i)} = \mathbf{b}^{(i,i)} = \mathbf{x}_i$ for $i = 1, \ldots, p$.

For $k = 1, \ldots, p-1$

For $i = 1, \ldots, p-k$

$$K_f^{(i,i+k)} = \frac{\mathbf{f}^{(i+1,i+k)^T}\mathbf{b}^{(i,i+k-1)}}{\mathbf{b}^{(i,i+k-1)^T}\mathbf{b}^{(i,i+k-1)}},$$

$$K_b^{(i,i+k)} = -\frac{\mathbf{f}^{(i+1,i+k)^T}\mathbf{b}^{(i,i+k-1)}}{\mathbf{f}^{(i+1,i+k)^T}\mathbf{f}^{(i+1,i+k)}},$$

$$\mathbf{f}^{(i,i+k)} = \mathbf{f}^{(i+1,i+k)} + K_f^{(i,i+k)}\mathbf{b}^{(i,i+k-1)},$$

$$\mathbf{b}^{(i,i+k)} = \mathbf{b}^{(i,i+k-1)} + K_b^{(i,i+k)}\mathbf{b}^{(i+1,i+k)}.$$

For $m = 1, \ldots, k-1$

$$\phi_m^{(i,i+k)} = \phi_m^{(i+1,I+k)} + K_f^{(i,i+k)}\beta_{k-m}^{(i,i+k-1)},$$

$$\beta_m^{(i,i+k)} = \beta_m^{(i,i+k-1)} + K_b^{(i,i+k)}\phi_{k-m}^{(i+1,i+k)},$$

$$\phi_k^{(i,i+k)} = K_f^{(i,i+k)},$$

$$\beta_k^{(i,i+k)} = K_b^{(i,i+k)}.$$

The complexity analysis of this algorithm is straightforward. In order to solve the problem determined by a pair (i, k) using solutions to the "smaller" problems $(i+1, i+k)$ and $(i, i+k-1)$, a total of $5n$ multiplications and 2 divisions are required, independently of the indices i and $k \geq 2$. Thus the operation count for finding the orthogonal columns of Q is obtained by summing over the number of such problems to be solved, giving

$$\begin{array}{ll} 5np(p-1)/2 & \text{multiplications} \\ p(p-1) & \text{divisions.} \end{array}$$

The computation of the coefficients is easily seen to involve about $p(p-1)(p-2)/3$ more multiplications. In cases where $n \gg p$, as is common in least-squares problems, the computational complexity is dominated roughly by a term of the form $5np^2/2$. By comparison, classical methods for orthogonalization have complexities of $cnp^2 +$ (lower order terms), where $c \leq 1$. Once again, it is important to note that the above algorithm actually computes QR factorizations of all contiguous subblocks of columns of the matrix X, so that comparisons of the complexity should seriously weigh this point.

In terms of the quantities computed in this algorithm, we have the factorization

$$X \begin{bmatrix} 1 & \phi_1^{(1,2)} & \phi_2^{(1,3)} & \cdots & \phi_{p-1}^{(1,p)} \\ 0 & 1 & \phi_1^{(1,3)} & & \cdot \\ & & & & \cdot \\ & & 1 & & \cdot \\ & 0 & & & \phi_1^{(1,p)} \\ & & & & 1 \end{bmatrix} = \left(\mathbf{f}^{(1,1)}, \mathbf{f}^{(1,2)}, \ldots, \mathbf{f}^{(1,p)}\right)$$

and the factorization

$$X \begin{bmatrix} 1 & 0 & & & 0 \\ \beta_1^{(1,p)} & & 1 & & \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & 1 & 0 \\ \beta_{p-1}^{(1,p)} & \cdot & \cdot \cdot & \beta_1^{(p-1,p)} & 1 \end{bmatrix} = \left(\mathbf{b}^{(1,p)}, \mathbf{b}^{(2,p)}, \ldots, \mathbf{b}^{(p,p)}\right).$$

In terms of space requirements, it can be readily seen that a total of $2n(p-1) + p(p-1)$ locations are sufficient for computing and storing the QR factorization. More location is required, of course, for keeping all the factorizations.

It is interesting to note that this algorithm has a simple and in fact, very tempting, recursive description which could be easily programmed. That is, the solution to a block of k contiguous columns is easily obtained from solutions to two subblocks of $k-1$ columns. Unfortunately, such a recursive algorithm would result in repeating the solution to a number of subproblems and thereby significantly increase the total operation count.

At this point we suggest an approach for using this algorithm to solve linear least-squares problems of the type

$$\min \| X\mathbf{a} - \mathbf{z} \|_2.$$

The described method explicitly computes a factorization

$$XT = Q,$$

where Q has orthogonal but not necessarily orthonormal columns. The matrix $P = Q(Q^TQ)^{-1/2}$ will have orthonormal columns, so the minimization is solved by

$$\mathbf{a} = T(Q^TQ)^{-1}Q^T\mathbf{z}. \tag{2.6}$$

Now one of the basic discoveries about solving least-squares problems by orthogonalization was that the computed Q in the QR decomposition is not guaranteed to have orthonormal columns and that one should not use the formula (2.6) for solving the problem (this is the case for Gram-Schmidt and modified Gram-Schmidt algorithms, but not for orthogonalization using Householder or Givens' transformations). In spite of this, it has been shown that it is possible to obtain an accurate solution by the modified Gram-Schmidt orthogonalization applied to the augmented matrix $(X, -\mathbf{z})$. The details of this are in [2] while a brief discussion and summary can be found in [18].

We suggest the same method for using the orthogonalization method of this paper in solving linear least-squares problems. That is, we suggest forming the augmented matrix $X' = (X, -\mathbf{z})$ and computing the factorization $X'T' = Q'$ as above. The last column of T' will then determine the least-squares solution to the original problem. Equivalently, we could adjoin $-\mathbf{z}$ to the right of X to form $(-\mathbf{z}, X)$ and then use the lower triangular decomposition to obtain the solution, since this factorization is explicitly computed also. An intriguing composition is to apply the described method to $(-\mathbf{z}, X, -\mathbf{z})$ and thereby obtain two explicit computed solutions and then combine them by say averaging. At present, we have no results or suggestions in this direction apart from these observations.

3. Orthogonalization of Matrices With Special Structure. Since the algorithm presented in this paper builds the QR factorization of a given matrix from QR factorizations of submatrices of that matrix, it is reasonable to expect that the algorithm simplifies if the submatrices were in some way related by some structure imposed on the full matrix. In the spirit of greatest generality, it will now be seen how the method simplifies if the columns of X are unitarily related. This yields an algorithm with the same structure as that given by De Meersman in [8], but we expect that our method has better stability properties. A further specialization is presented in the next section, showing that this method also gives the well-known Itakura-Saito lattice algorithm for linear prediction.

Suppose that X has columns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ with $\mathbf{x}_j = U^{j-1}\mathbf{x}_1$, where U is an orthogonal transformation on R^n. This structure imposes the following simplification on the solution to the subproblems:

LEMMA. *Let X be a matrix as described above. Let* $\mathbf{r}$ *and* $\mathbf{s}$ *be k-vectors which are solutions to*

$$\| X(j, j + k - 1)\mathbf{r} + \mathbf{x}_{j+k} \|_2 = \textit{minimum}, \tag{3.1}$$

$$\| X(j + 1, j + k)\mathbf{s} + \mathbf{x}_j \|_2 = \textit{minimum}. \tag{3.2}$$

Then $\mathbf{r}$ *and* $\mathbf{s}$ *are independent of* j *and, furthermore, if*

$$\mathbf{r} = (r_1, \ldots, r_k)^T, \qquad \mathbf{s} = (s_1, \ldots, s_k)^T,$$

then in fact $r_i = s_{k-i+1}$.

Proof. We begin by noting that

$$N = X(j, j+k-1)^T X(j, j+k-1) = X(j+1, j+k)^T X(j+1, j+k)$$

and that these matrices are independent of j. Furthermore, these matrices are Toeplitz, symmetric and positive-definite. These facts follow from the observation that for orthogonal U we have

$$\mathbf{x}_i^T \mathbf{x}_m = \mathbf{x}_1^T (U^{i-1})^T U^{m-1} \mathbf{x}_1 = \mathbf{x}_1 U^{i-m} \mathbf{x}_1 = \mathbf{x}_1 U^{m-i} \mathbf{x}_1.$$

Now if E is the k by k exchange matrix, that is

$$E = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ & \iddots & & \\ 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

then we have similarly

$$\mathbf{c} = X(j, j+k-1)^T \mathbf{x}_{j+k} = EX(j+1, j+k)^T \mathbf{x}_j.$$

Thus the normal equations for (3.1) and (3.2) are respectively

$$N\mathbf{r} = \mathbf{c} \quad \text{and} \quad N\mathbf{s} = E\mathbf{c}.$$

Since $E^2 = I_k$, the k by k identity, and $ENE = N$ for Toeplitz N, we actually have that s satisfies

$$EN\mathbf{s} = ENEE\mathbf{s} = NE\mathbf{s} = E^2\mathbf{c} = \mathbf{c},$$

so that $E\mathbf{s} = \mathbf{r}$ which is precisely the final claim of the lemma. □

One of the immediate consequences of this lemma is that

$$K_f^{(i,i+k)} = K_b^{(i,i+k)} = K_f^{(1,1+k)} = K_k,$$

where these are the quantities occurring in the algorithm of Section 2, and from the definition of these quantities it follows that

$$\begin{aligned} \mathbf{f}^{(i,i+k)^T} \mathbf{f}^{(i,i+k)} &= \mathbf{f}^{(1,1+k)^T} \mathbf{f}^{(1,1+k)} \\ &= \mathbf{b}^{(i,i+k)^T} \mathbf{b}^{(i,i+k)} = \mathbf{b}^{(1,1+k)^T} \mathbf{b}^{(1,1+k)}, \end{aligned} \tag{3.3}$$

which is a relation that can be used to reduce the multiplication count. There is however substantial evidence [6] indicating that a more stable version of this algorithm uses the denominator

$$\sqrt{\mathbf{f}^{(1,1+k)^T} \mathbf{f}^{(1,1+k)} \mathbf{b}^{(1,1+k)^T} \mathbf{b}^{(1,1+k)}}$$

in the computation of the coefficient K_k.

Furthermore, we note that

$$\mathbf{f}^{(i,i+k)} = U^{i-1} \mathbf{f}^{(1,1+k)} = U^{i-1} \mathbf{f}_k \quad \text{and} \quad \mathbf{b}^{(i,i+k)} = U^{i-1} \mathbf{b}^{(1,1+k)} = U^{i-1} \mathbf{b}_k,$$

as a simple calculation shows.

Clearly all of the required inner products involving the residuals, $\mathbf{f}^{(i,i+k)}$ and $\mathbf{b}^{(i,i+k)}$, are computable from the basic residuals $\mathbf{f}_k$ and $\mathbf{b}_k$ according to

$$\mathbf{f}^{(i+1,i+k)^T}\mathbf{b}^{(i,i+k-1)} = \left(U^i\mathbf{f}_{k-1}\right)^T U^{i-1}\mathbf{b}_{k-1} = \mathbf{f}_{k-1}^T U^{-1}\mathbf{b}_{k-1}.$$

The final columns of $Q = (\mathbf{q}_1,\ldots,\mathbf{q}_p)$ are then

$$\mathbf{q}_j = \mathbf{f}^{(1,j)} = \mathbf{f}_{j-1},$$

where we define $\mathbf{f}^{(1,1)} = \mathbf{x}_1 = \mathbf{f}_0 = \mathbf{b}_0 = \mathbf{q}_1$.

Hence the algorithm for computing Q and T for such a structured X may be summarized as follows:

Initialization: $\mathbf{x}_1 = \mathbf{f}_0 = \mathbf{b}_0$

For $k = 1,\ldots,p-1$ do

$$K_k = -\frac{\left(U\mathbf{f}_{k-1}\right)^T\mathbf{b}_{k-1}}{\left(\mathbf{b}_{k-1}^T\mathbf{b}_{k-1}\mathbf{f}_{k-1}^T\mathbf{f}_{k-1}\right)^{1/2}},$$

$$\mathbf{f}_k = U\mathbf{f}_{k-1} + K_k\mathbf{b}_{k-1},$$

$$\mathbf{b}_k = \mathbf{b}_{k-1} + K_kU\mathbf{f}_{k-1}.$$

For $k = 1,\ldots,p-1$

For $i = 1,\ldots,k-1$

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

$$a_k^k = K_k.$$

Then with $Q = (\mathbf{f}_0,\mathbf{f}_1,\ldots,\mathbf{f}_{p-1})$ and

$$T = \begin{bmatrix} 1 & a_1^1 & a_2^2 & \cdot & \cdot & \cdot & a_{p-1}^{p-1} \\ 0 & 1 & a_1^2 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & & & & \\ & & & \ddots & & 1 & a_1^{p-1} \\ 0 & & & & 0 & 1 & \end{bmatrix},$$

the desired factorization is obtained.

A similar algorithm was derived in [8] but using a less general approach. The above method has more promising stability properties as will be discussed in the next section.

In the complexity analysis of this specialized algorithm it is assumed that the first column $\mathbf{x}_1$ of X and the orthogonal U are the only quantities given at the start, and therefore the vectors $U\mathbf{f}_k$ need to be computed as they are required by the algorithm. In general an evaluation of $U\mathbf{f}$ will require n^2 multiplications, but we shall simply assume that $U\mathbf{f}$ requires $m(n)$ multiplications. Thus the operation count is

$m(n)(p-1) + 5n(p-1)$	multiplications,
$p-1$	divisions,
$p-1$	square roots.

By comparison, the computation of the columns of X one by one from $\mathbf{x}_1$ and U would alone require $m(n)\,(p-1)$ multiplications. If a general purpose algorithm

for computing the QR factorization is then used, it would involve another $O(np^2)$ operations so that savings are felt only in lower order terms. In the next section, the case where U is the circular shift operator is discussed in which case $m(n) = 0$ since U involves reindexing only. We postpone discussion of least-squares problems of this type to the next section where this particular structure is commonly used in applications and is usually referred to as Wiener filtering.

4. Applications to Time Series Prediction. The most important application of this specialized structure and specialized algorithm is fortunately a case where U is extremely simple. If U is the circular shift operator defined by

$$U\mathbf{x} = (s_{i-1(\text{mod } n)}), \qquad \mathbf{x} = (s_i),$$

then the computation of $U\mathbf{x}$ involves no arithmetic computations but only reindexing of the arrays, and the matrix X is circulant vertically. In that case the algorithm requires only

$$5n(p-1) \text{ multiples and } p-1 \text{ divisions}$$

to compute Q and the K_k coefficients. The computation of the entries of T involves

$$1/2\,(p-1)(p-2) \text{ multiplies.}$$

The situation where U is the cyclical shift operator is precisely the situation encountered in the linear prediction of stationary time series. Let (s_i) be a time series with only finitely many nonzero terms, say $s_i \neq 0$ for $1 < i < n$. Then the linear prediction problem of order p for the series (s_i) is to find the coefficients $a_1,\ldots,a_p$ which minimize the mean-squared prediction error

$$E_p = \sum_i (s_i + a_1\mathbf{s}_{i-1} + \cdots + a_p\mathbf{s}_{i-p})^2. \tag{4.1}$$

This may be formulated as a matrix least-squares problem by writing $X' = (\mathbf{x}_1,\ldots,\mathbf{x}_p)$

$$\mathbf{x}_0 = (s_i), \qquad i = 1,\ldots,n+p,$$
$$\mathbf{x}_i = U^i\mathbf{x}_0, \qquad i = 1,\ldots,p.$$

This special form is clearly circulant, but, furthermore, has zero boundaries in the obvious sense. A descriptive name for such a structure would therefore be "Toeplitz with zero boundaries." The problem then becomes

$$\text{minimize } E_p = \| X'\mathbf{a} - \mathbf{x}_0 \|_2^2. \tag{4.2}$$

By the previous comments the desired coefficients $a_1,\ldots,a_p$ appear in the last column of the unit upper triangular matrix T defined by the condition that XT have orthogonal columns, where $X = (\mathbf{x}_0, \mathbf{x}_1,\ldots,\mathbf{x}_p) = (\mathrm{x}_0,\ X')$ and

$$T = \begin{bmatrix} 1 & a_{1,1} & a_{2,2} & \cdot & \cdot & \cdot & a_p \\ 0 & 1 & a_{2,1} & \cdot & \cdot & \cdot & a_{p-1} \\ 0 & & & & & 1 & a_1 \\ 0 & 0 & & & & & 1 \end{bmatrix}.$$

The algorithm for computing XT and T is then:

The Lattice Algorithm [4], [8], [12].

Initialization: $\mathbf{f}_0 = \mathbf{b}_0 = \mathbf{x}_0$.

For $i = 1, \ldots, p$ do

$$K_i = -\frac{\Sigma_j f_{i-1,j-1} b_{i-1,j}}{\left(\Sigma_j b_{i-1,j}^2 \Sigma_j f_{i-1,j}^2\right)^{1/2}}.$$

For $j = 1, \ldots, n + p$ do

$$f_{i,j+1} = f_{i-1,j} + K_i b_{i-1,j+1},$$
$$b_{i,j+1} = K_i f_{i-1,j} + b_{i-1,j+1},$$

$f_{i,1} = K_i b_{i-1,1}$,

$b_{i,1} = b_{i-1,1}$.

For $i = 1, \ldots, p$ do

For $j = 1, \ldots, i - 1$ do

$$a_{i,j} = a_{i-1,j} + K_i a_{i-1,i-j},$$

$a_{i,i} = K_i$.

At the end of the computation

$$T = \begin{bmatrix} 1 & a_{1,1} & a_{2,2} & \cdots & a_{p,p} \\ 0 & 1 & a_{2,1} & \cdots & a_{p,p-1} \\ 0 & 0 & 1 & & \cdot \\ \cdot & & & \diagdown & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & a_{p,1} \\ 0 & \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

with $a_i = a_{p,i}$ as desired. Furthermore,

$$XT = (\mathbf{f}_0, \ldots, \mathbf{f}_p) = Q \tag{4.3}$$

has orthogonal columns.

It is interesting to note that the Itakura-Saito-Burg algorithm has its historical roots in the construction of difference schemes and continuity conditions for the one-dimensional wave equation in a nonhomogeneous medium. (The name "lattice" comes from the algorithm's flowgraph which resembles a lattice or ladder.)

The projection coefficients K_I are precisely the partial correlation coefficients and as such are extremely significant in testing hypotheses concerning the order, p, of the autoregressive process fitting the series $\{s(i)\}$ [3], [16]. In some applications, such as data compression for speech transmission [17] only the K_I coefficients are used. Note that from (3.3), and the Cauchy-Schwartz inequality, it follows that $|K_I| < 1$. (The strict inequality follows from the fact that X has full rank p.)

Furthermore, since

$$\|\mathbf{f}_I\|^2 = \|\mathbf{b}_I\|^2 = E_I,$$

it follows that

$$E_{I+1} = \|\mathbf{f}_{I+1}\|^2 = \|U\mathbf{f}_I + K_{I+1}\mathbf{b}_I\|^2 \tag{4.4}$$
$$= \|U\mathbf{f}_I\|^2 + 2K_{I+1}(U\mathbf{f}_I)^T\mathbf{b}_I + K_{I+1}\|\mathbf{b}_I\|^2$$
$$= \|\mathbf{f}_I\|^2(1 - K_{I-1}^2) = E_I(1 - K_{I+1}^2)$$

by (3.3) and (4.2).

Thus E_p has a simple recursive form convenient for computation:

$$E_I = E_{I-1}(1 - K_I^2).$$

A classical result [11] is that

$$\det(X^TX) = \prod_{I=1}^{p} E_I$$

and that the eigenvalues of X^TX are contained in the range of the power spectral density of the series $\{s(i)\}$ so that the singular values of X are contained in the square root of the range of this power spectral density. Furthermore, as was shown in [5], the 2-condition number of X^TX satisfies

$$K_2(X^TX) \leqslant p\frac{\prod_{i=1}^{p}(1 + |K_i|)}{\prod_{i=1}^{p}(1 + |K_i|)} = C,$$

so that the 2-condition number of X satisfies

$$K_2(X) \leqslant \sqrt{C}. \tag{4.5}$$

It is well known that least-squares problems are better solved by orthogonalization than by passing to covariance matrices [10], since the error analysis of the former reflects closer the perturbation theory of the least-squares problem itself [18]. This is a point much overlooked in the arguments for using this lattice method. Most arguments center on the method's computational convenience and robustness [14], while the accuracy properties are extremely good both empirically [15] and analytically [6].

Solutions by solving the covariance normal equations are usually discouraged [3], and it is evident that this lattice method offers an alternative.

By way of an error analysis, it can be shown that if Q is the computed version of $\bar{Q}$ in (4.3), then

$$\|Q - \bar{Q}\|_2 \leqslant np^{1/2}\sqrt{C} \tag{4.6}$$

which compares favorably with the condition number bound (4.5). Inequality (4.6) holds only for the algorithm in which the partial correlation coefficients K_i are computed according to the form in the Lattice Algorithm, even though (3.3) suggests there are algebraically equivalent forms which involve fewer computations and no square roots. If the square root formulation is not used, the forward stability as expressed by (4.6) is lost [6].

If s_i is replaced by some other series y_i so that the object is to minimize

$$E_p = \sum_i (y_i + b_1 s_{i-1} + \cdots + b_p s_{i-p})^2,$$

the problem is called the discrete-time Wiener filtering problem. This least-squares problem can be solved exactly as in Section 2, that is, by adjoining the vector

$$\mathbf{y} = (y_i)$$

to matrix X'. In this way, the structure of the problem can be exploited in the solution of all subproblems except for subproblems involving the column $\mathbf{y}$. This approach would result in an operation count with a leading term of $8np$.

5. Summary. This paper has presented an orthogonalization technique based on projections which computes orthogonal factorizations of all contiguous subblocks of columns of the matrix. The algorithm uses fewer computations than required by updating factorizations otherwise. This general orthogonalization procedure simplifies significantly when the underlying matrix has a circulant structure, a structure important in time series and engineering applications. The streamlined algorithm thus obtained is the known "Lattice Method" for linear prediction. This shows that the Lattice Method is a legitimate orthogonalization algorithm for solving the least-squares problem arising in linear prediction of time series. Together with the previous work of the author in [5], this shows that for small residual linear prediction problems, the lattice algorithm is to be much preferred over solving the covariance equations since the condition number of the underlying matrix is on the order of the reciprocal of the residual sum of squares.

Acknowledgements. The author wishes to thank V. Klema of MIT and Gene Golub of Stanford for facilities and discussions used in this work. Sincere appreciation goes to C. Van Loan and the anonymous referees for their suggestions and comments which have much improved the presentation of this paper.

Department of Mathematics
Tufts University
Medford, Massachusetts 02155

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

1. D. A. BELSLEY, E. KUH & R. E. WELSCH, *Regression Diagnostics*, Wiley, New York, 1980.
2. Å. BJÖRCK, "Solving linear least squares problems by Gram-Schmidt orthogonalization," *BIT*, v. 7, 1967, pp. 1–21.
3. G. E. BOX & G. M JENKINS, *Time Series Forcasting and Control*, Holden-Day, San Francisco, 1970.
4. J. P. BURG, *Maximal Entropy Spectral Analysis*, Ph.D. dissertation, Stanford University, 1975.
5. G. CYBENKO, "The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations," *SIAM J. Sci. Statist. Comput.*, v. 1, 1980, pp. 303–320.
6. G. CYBENKO, *Rounding-Errors and Nonoptimality of Lattice Methods for Linear Prediction*, Proc. 14th Annual Princeton Conference on Information Systems and Sciences, March 1980.
7. P. DELSARTE, Y. GENIN & Y. KEMP, "A method of matrix inverse triangular decomposition based on contiguous principal submatrices," *Linear Algebra Appl.* (To appear.)
8. R. DE MEERSMAN, "A method for least squares solutions of systems with a cyclic coefficient matrix," *J. Comput. Math.*, v. 1, 1975, pp. 51–54.
9. G. M. FURNIVAL & R. W. WILSON, JR., "Regression by leaps and bounds," *Technometrics*, v. 16, 1974, pp. 499–511.
10. G. H. GOLUB, "Numerical methods for solving least squares problems," *Numer. Math.*, v. 7, 1965, pp. 206–216.
11. U. GRENANDER & G. SZEGO, *Toeplitz Forms and their Applications*, Univ. of California Press, Berkeley, 1958, p. 38.

12. F. ITAKURA & S. SAITO, *Digital Filtering Techniques for Speech Analysis and Synthesis*, Proc. 7th Internat. Congr. Acoust., Budapest, 1971, pp. 261–264.

13. T. KAILATH, "A view of three decades of linear filtering theory," *IEEE Trans. Information Theory*, v. 2, 1974, pp. 146–181.

14. J. MAKHOUL, "A class of all-zero lattice digital filters: properties and applications," *IEEE Trans. Acoust. Speech Signal Process.*, v. 4, 1978, pp. 304–314.

15. J. MAKHOUL, personal communication, 1979.

M. H. QUENOUILLE, "The joint distribution of serial correlation coefficients,"*Ann. Math. Statist.*, v. 20, 1949, pp. 561–571.

17. L. R. RABINER & R. SCHAFER, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, N.J., 1978.

18. G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

19. J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, London and New York, 1965.