PRESIDENCY OF THE REPUBLIC

THE INSTITUTE OF NATIONAL PLANNING

التخطيط القومى

Memo. No. 85

A NON-RECURSIVE METHOD OF MATRIX INVERSION AND ITS APPLICATIONS TO REGRESSION ANALYSIS.

by

Dr. M. M. El-Imam

14th November, 1961.

I. INTRODUCTION

The purpose of this paper is to reformulate the compact single-division method [1] in such a manner as to obtain non-recursive formulae for the backward solution. The advantages of the method are illustrated through certain applications.

In the classical method, the matrix to be inverted, $\underline{M}$ say[2] is transformed into two triangular matrices $\underline{X}$ & $\underline{Y}$ such that:

$$\underline{M} = \underline{X} \cdot \underline{Y} \qquad (1)$$

where $\underline{X}$ has zeros above the principal diagonal, and $\underline{Y}$ has zeros below it. At the same time, the inverse of $\underline{X}$ is obtained, say $\underline{Z}=\underline{X}^{-1}$ This constitutes what is called the "forward solution".

The inverce of M is:

$$\underline{M}^{-1} = \underline{Y}^{-1}\underline{X}^{-1} = \underline{Y}^{-1}\underline{Z} \qquad (2)$$

This latter formula requires in fact two steps: the computation of $\underline{Y}^{-1}$ and the multiplication of $\underline{Y}^{-1}$ into $\underline{Z}$. The classical compact method achieves this by carrying out the two steps simultaneously, through what is called the "backward solution". This might have the advantage of giving the inverse in the most concise number of steps. However it has some disadvantages, especially from the procedural point of view. First, the rules of computation are distinctly different between the forward and backward solutions. Further the positions of rows and columns to be used together in the backward solution are wide apart. These tend to make it extremely difficult especially for ordinary computers. One might add also that the necessary rules for checking are rather complicated.

On the other hand, the backward solution makes use of the rows (or columns) already obtained in the inverse to build up the rest. This has the disadvantage of carrying over the errors of rounding off throughout. Further in order to obtain a specific element in the inverse one has to compute all the subsequent ones (with respect to order). This makes the method rather expensive and time-consuming when only certain elements of the inverse matrix are required, the diagonal ones only for example.

In some cases one might need also the inverses of certain submatrices of $\underline{M}$. For example, in the multiplex method of linear programming, there is a need to obtain the inverses of matrices of all consecutive orders starting from the first. As we have already noticed, the classical method starts its backward solution from the

(1) Cf.. Dwyer : Linear Computations, p. 103 - See also, R. Frisch: "The compact method for solving linear equations and inverting matrices in the non-symmetric case", Memo. no. 1, N.P.C. - Cairo, 24/11/1957

(2) We shall use underlined letters to denote vectors and matrices, denoting matrices by capital letters and vectors by small ones.

last row (or column). This means that the results of any operation will not be useful for the rest. Of course methods of reduction of the order of the inverted matrix, and methods of building up are available.[3] However, these methods are rather cumbersome, especially in the process of calculating the successive moment matrices inverses as in the multiplex method.

What we propose here is the following:

Since in the so-called "forward" solution we obtain $\underline{X}$, $\underline{Y}$ & $\underline{X}^{-1}$, we can in the meantime obtain $\underline{Y}^{-1}$ also. This will not involve much work since $\underline{Y}$, like $\underline{X}$, is triangular. The inverse $\underline{M}^{-1}$ can be then computed as the product of two triangular matrices, as defined in (2). As will be shown later this will help to overcome the above-mentioned difficulties. In particular this method avoids the recursive formulae of the backward solution. It helps further in the process of building-up the inverses of submatrices in a simultaneous manner. This is due to the fact that the building-up (or down) of the inverse of a triangular matrix is quite simple. The method is self-checking in a consistent and comprehensive manner. We shall begin by considering the asymmetric case, noticing that the method is most suitable for the symmetric case where it does not involve any extra computations. The symmetric case can be obtained as a special case. Applications where the method gives most suitable results are given later.

II - THE ASYMMETRIC CASE

Let $\underline{M}$ be the real, non-singular square asymmetric matrix to be inverted. For purposes of computation we border it by two unit matrices: one on the right-hand side, and the other below, each being of the same order as $\underline{M}$, viz., n.n. Putting N = 2 n, then we obtain the N.N. matrix:

$$\underline{A} = \begin{bmatrix} \underline{M} & \underline{I} \\ \underline{I} & \underline{0} \end{bmatrix} \qquad (3)$$

whose south-east submatrix is empty. This matrix is to be written on the top of the sheet, together with two extra columns to its right. The first of these stands for "row sums":

$$a_{i.} = \sum_{j=1}^{N} a_{ij} = \sum_{j=1}^{n} a_{ij} + 1 \qquad (i=1,\dots n) \qquad (4)$$

The other column is for "checks":

$$\sum_{j=1}^{n} m_{ij} + 1 \qquad (= a_{i.}) \qquad (i=1,\dots,n) \qquad (5)$$

where m_{ij} $(=a_{ij}$, for i, j=1,...n) are read from the original matrix, $\underline{M}$. No checks are required for the remaining (unit) rows. Below $\underline{A}$ we introduce a row for column sums:

$$a_{.j} = \sum_{i=1}^{n} a_{ij} \qquad (j=1,\dots,n) \qquad (6)$$

(3) Frisch, op. cit.

which can be checked against $(\sum_{i=1}^{n} m_{ij} + 1)$.

A new matrix $\underline{D}$, of order N.N. is obtained and registered below $\underline{A}$ in the following manner. On and below the principal diagonal, in the first n columns, we build up successively the following elements:

$$c_{ij} = a_{ij} - \sum_{k=1}^{j-1} c_{ik} b_{kj} \qquad \begin{pmatrix} i = j, j+1, \ldots, N \\ j = 1, \ldots\ldots, n \end{pmatrix} \quad (7)$$

The elements b_{kj} are those above the principal diagonal of $\underline{D}$, in the first n rows, and are obtained as follows:

$$b_{kj} = c_{kj}/c_{kk}, \qquad c_{kj} = a_{kj} - \sum_{h=1}^{k-1} c_{kh} b_{hj} \quad \begin{pmatrix} k=1,2,\ldots,n \\ j=k+1,\ldots,N \end{pmatrix} \quad (8)$$

Thus c_{kj} are obtained in the same manner as c_{ij} in (7), and they are transformed into b_{kj}, by a simple division operation. For computational purposes, one usually registers the c_{kj} in an additional sheet subdivided in the same columns as $\underline{D}$, and checked directly then multiplied into a constant scalar = $1/c_{kk}$, the products being registered in the corresponding positions in $\underline{D}$ as b_{kj}. This covers all elements in $\underline{D}$ except those in the rows and columns nos.: n+1,...,N. These latter (corresponding to the null matrix in A) are the elements of the inverse $\underline{K} = \underline{M}^{-1}$, and they are obtained as follows:

$$k_{ij} = d_{n+i,n+j} = \sum_{h=1}^{n} c_{n+i,h} b_{h,n+j} \qquad (i,j=1,\ldots,n) \quad (9)$$

For checking purposes, we have first the row sums:

$$b_{i.} = \sum_{j=i+1}^{n} b_{ij} \qquad (i=1,\ldots\ldots,n) \quad (10)$$

Then we check by:

$$a_{i.} - \sum_{h=1}^{i-1} c_{ih} b_{h.} - c_{ii} b_{i.} = c_{ii} \qquad (i=1,\ldots,n) \quad (11)$$

Again we obtain the column sums and write them below $\underline{D}$. It is advisable here to build the sums upwards, in order to obtain the sub-sums:

$$f_{.j} = \sum_{i=N}^{n+1} c_{ij} \qquad (j=1,\ldots,n) \quad (12)$$

which can be written down immediately and below them we write the complete sums obtained by continuing the same operation:

$$c_{.j} = f_{.j} + \sum_{i=n}^{j} c_{ij} = \sum_{i=j}^{N} c_{ij} \qquad (j=1,\ldots\ldots,n) \quad (13)$$

Both sums are checked simultaneously (apart from errors in copying down $f_{.j}$) by

$$a_{.j} - \sum_{k=1}^{j-1} c_{.k} b_{kj} - c_{.j} = 0 \quad (14)$$

Finally, the check on the elements of the inverse in made through checking its column as follows:

$$k_{.j} = \sum_{i=1}^{n} k_{ij}, \qquad \sum_{h=1}^{n} f_{.h} b_{hj} - k_{.j} = 0 \quad (15)$$

Similar checks for rwos can be easily obtained in an obvious manner.

Our results can be summarised in the following proposition:

PROPOSITION:

For the matrix $\underline{M}^{-1}$, the triangular matrices $\underline{X}^{-1}$ and $\underline{Y}^{-1}$ defined in (1) and (2) are:

$$\underline{X}^{-1} = \begin{bmatrix} b_{1,n+1} & 0 & 0 & \cdots & 0 \\ b_{2,n+1} & b_{2,n+2} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b_{n,n+1} & b_{n,n+2} & b_{n,n+3} & \cdots & b_{n,N} \end{bmatrix} \quad (16)$$

and

$$\underline{Y}^{-1} = \begin{bmatrix} c_{n+1,1} & c_{n+1,2} & c_{n+1,3} & \cdots & c_{n+1,n} \\ 0 & c_{n+2,2} & c_{n+2,3} & \cdots & c_{n+2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & c_{N,n} \end{bmatrix} \quad (17)$$

Further,

$$\underline{X} = \underline{M}.\underline{Y}^{-1} = \begin{bmatrix} c_{11} & 0 & \cdots & 0 \\ c_{21} & c_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \quad (18)$$

Similarly,

$$\underline{Y} = \underline{X}^{-1}\underline{M} = \begin{bmatrix} 1 & b_{12} & \cdots & b_{1n} \\ 0 & 1 & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (19)$$

These relations show the main differences between the familiar method and ours. Both methods involve the evaluation of $\underline{X}$ and $\underline{Y}$ whose product is, by (1), equal to $\underline{M}$. In our method, both the two inverses $\underline{X}^{-1}$ and $\underline{Y}^{-1}$ are separately calculated, and their product is then obtained to give $\underline{M}^{-1}$. In the classical method either of the two inverses is calculated. The inverse of the other is obtained in a recursive manner together with its product into the other to give $\underline{M}^{-1}$ simultaneously. The recursiveness is due to the diagonality of $\underline{X}$ and $\underline{Y}$.

Table (1) shows a schematic representation of the steps involved in the process of computation as suggested in this proposition for the case where $n = 4$.

Table (1)

A Schematic Representation of the Inversion of a Matrix of the Fourth-Order

(Part 1 "Left")

Order	1	2	3	4
1	a_{11}	a_{12}	a_{13}	a_{14}
2	a_{21}	a_{22}	a_{23}	a_{24}
3	a_{31}	a_{32}	a_{33}	a_{34}
4	a_{41}	a_{42}	a_{43}	a_{44}
5	1.0	0	0	0
6	0	1.0	0	0
7	0	0	1.0	0
8	0	0	0	1.0
Σ	$a_{.1}=\sum_1^8 a_{ij}$	$a_{.2}=\sum_1^8 a_{j2}$	$a_{.3}=\sum_1^8 a_{j3}$	$a_{.4}=\sum_1^8 a_{j4}$
1	$c_{11}= a_{11}$	$b_{12}=c_{12}/c_{11}$	$b_{13}=c_{13}/c_{11}$	$b_{14}=c_{14}/c_{11}$
2	$c_{21}= a_{21}$	$c_{22}=a_{22}-c_{21}b_{12}$	$b_{23}=c_{23}/c_{22}$	$b_{24}=c_{24}/c_{22}$
3	$c_{31}= a_{31}$	$c_{32}=a_{32}-c_{31}b_{12}$	$c_{33}=a_{33}-\sum_1^2 c_{3i}b_{i3}$	$b_{34}=c_{34}/c_{33}$
4	$c_{41}= a_{41}$	$c_{42}=a_{42}-c_{41}b_{12}$	$c_{43}=a_{43}-\sum_1^2 c_{4i}b_{i3}$	$c_{44}=a_{44}-\sum_1^3 c_{4i}b_{i4}$
5	$c_{51}=1.0$	$c_{52}= -b_{12}$	$c_{53}= -\sum_1^2 c_{5i}b_{i3}$	$c_{54}=- c_{5i}b_{i4}$
	0	$c_{62}=1.0$	$c_{63}= -b_{23}$	$c_{64}=- c_{6i}b_{i4}$
	0	0	$c_{73}=1.0$	$c_{74}=- b_{34}$
	0	0	0	$c_{84}=1.0$
Σ_1	$f_{.1}=\sum_5^8 c_{i1}$	$f_{.2}=\sum_5^8 c_{i2}$	$f_{.3}=\sum_5^8 c_{i3}$	$f_{.4}=\sum_5^8 c_{i4}$
Σ	$c_{.1}=f_{.1}+\sum_1^4 c_{i1}$	$c_{.2}=f_{.2}+\sum_2^4 c_{i2}$	$c_{.3}=f_{.3}+\sum_3^4 c_{i3}$	$c_{.4}=f_{.4}+c_{44}$
Check	$a_{.1}$	$a_{.2}-c_{.1}b_{12}$	$a_{.3}-\sum_1^2 c_{.i}b_{i3}$	$a_{.4}-\sum_1^2 c_{.1}b_{i4}$

Auxiliary Table

1	$c_{11}=a_{11}$	$c_{12}=a_{12}$	$c_{13}=a_{13}$	$c_{14}=a_{14}$
2	–	$c_{22}=a_{22}-c_{21}b_{12}$	$c_{23}=a_{23}-c_{21}b_{13}$	$c_{24}=a_{24}-c_{21}b_{14}$
3	–	–	$c_{33}=a_{33}-\sum_1^2 c_{3i}b_{i3}$	$c_{34}=a_{34}-\sum_1^2 c_{3i}b_{i4}$
4	–	–	–	$c_{44}=a_{44}-\sum_1^3 c_{4i}b_{i4}$

Table (1), cont.

(Part II)
Right

	5	6	7	8	Check
1	1.0	0	0	0	$a_{1.}$
2	0	1.0	0	0	$a_{2.}$
3	0	0	1.0	0	$a_{3.}$
4	0	0	0	1.0	$a_{4.}$
5	0	0	0	0	
6	0	0	0	0	
7	0	0	0	0	
8	0	0	0	0	
Σ					
1	$b_{15}=1/c_{11}$	0	0	0	$c_{1.}/c_{11}$
2	$b_{25}=c_{25}/c_{22}$	$b_{26}=1/c_{22}$	0	0	$c_{2.}/c_{22}$
3	$b_{35}=c_{35}/c_{33}$	$b_{36}=c_{36}/c_{33}$	$b_{37}=1/c_{33}$	0	$c_{3.}/c_{33}$
4	$b_{45}=c_{45}/c_{44}$	$b_{46}=c_{46}/c_{44}$	$b_{47}=c_{47}/c_{44}$	$b_{48}=1/c_{44}$	$c_{4.}/c_{44}$
5	$k_{11}=\sum_1^4 c_{5i}b_{i5}$	$k_{12}=\sum_2^4 c_{5i}b_{i6}$	$k_{13}=\sum_3^4 c_{5i}b_{i7}$	$k_{14}=c_{54}b_{48}$	$\sum_1^4 c_{5i}b_{i.}$
6	$k_{21}=\sum_2^4 c_{6i}b_{i5}$	$k_{22}=\sum_2^4 c_{6i}b_{i6}$	$k_{23}=\sum_3^4 c_{6i}b_{i7}$	$k_{24}=c_{64}b_{48}$	$\sum_2^4 c_{6i}b_{i.}$
7	$k_{31}=\sum_3^4 c_{7i}b_{i5}$	$k_{32}=\sum_3^4 c_{7i}b_{i6}$	$k_{33}=\sum_3^4 c_{7i}b_{i7}$	$k_{34}=c_{34}b_{48}$	$\sum_3^4 c_{7i}b_{i.}$
8	$k_{41}=b_{45}$	$k_{42}=b_{46}$	$k_{43}=b_{47}$	$k_{44}=b_{48}$	$b_{4.}$
Σ_j	$k_{.1}=\sum_1^4 k_{i1}$	$k_{.2}=\sum_1^4 k_{i2}$	$k_{.3}=\sum_1^4 k_{i3}$	$k_{.4}=\sum_1^4 k_{i4}$	
Σ					
Check	$\sum_1^4 f_{.i}b_{i5}$	$\sum_2^4 f_{.i}b_{i6}$	$\sum_3^4 f_{.i}b_{i7}$	$f_{.4}b_{48}$	

Auxiliary Table, cont.

1	$c_{15}=1.0$	0	0	0	$a'_{1.}$
2	$c_{25}=-c_{21}b_{15}$	$c_{26}=1.0$	0	0	$a_{2.}-c_{21}b_{1.}$
3	$c_{35}=-\sum_{1}^{2}c_{3i}b_{i5}$	$c_{36}=-c_{32}b_{26}$	$c_{37}=1.0$	0	$a_{3.}-\sum_{1}^{2}c_{3i}b_{i.}$
4	$c_{45}=-\sum_{1}^{3}c_{4i}b_{i5}$	$c_{46}=-\sum_{1}^{3}c_{4i}b_{i6}$	$c_{47}=-c_{43}b_{37}$	$c_{48}=1.0$	$a_{4.}-\sum_{1}^{3}c_{4i}b_{i.}$

In the actual process of computation, one starts with the first column of c's, and the first row in the auxiliary table, from which the first row of b's is obtained. Then we calculate the second column of c's, and the second row in the auxiliary table, from which the second row of b's follows. Similarly for the third, then the fourth columns and rows.

A slight variation on the order of the table can be made as follows. First, the unit matrix below the $\underline{M}$ matrix (and the null matrix beside it) can be dropped to spare space without affecting the results, since we bear in mind that it is there. Futher, we can insert the elements of the auxiliary table in the main table in place of the corresponding b's. This will help to retain the same type of operation both row-wise and column-wise. The corresponding b's would be then written in a transposed manner on another sheet. Using this sheet by folding it so as to have the required b's apparent and putting the column thus exposed beside the relevant c-column, we can obtain the products required for calculating the subsequent rows and columns of the c's. For obtaining the inverse, i.e., the k's, we can use the relavant b's in the same manner. This would relieve the computer's eye from moving row-wise and column-wise at the same time. But the underlying formulae and method will remain unchanged.

III. THE SYMMETRIC CASE

It is clear that the general formulae given in the previous section can be directly applied to the case of inverting symmetric matrices. The symmetry introduces a lot of simplifications which can be summarised as follows. First, it would be natural to write $\underline{A}$ simply as ($\underline{M}$ $\underline{I}$). Further we can neglect the elements lying on one side of the principal diagonal, e.g., those below it. Thus:

$$a_{i.} = \sum_{j=1}^{i-1} a_{ji} + \sum_{j=1}^{n} a_{ij} + 1 \qquad (i=1,\ldots,n) \qquad (20)$$

and similarly for the check (5). There will be no need to add the two rows of checks and sums defined by (6).

The matrix $\underline{D}$ is calculated as follows : the c_{ij} are calculated as in (7), using the property of the symmetry, viz.,

$$c_{ij} = c_{ji}$$

i.e., we can use (8) instead. It follows that the elements b_{kj} can be obtained by:

$$b_{kj} = c_{jk}/c_{jj} \qquad \begin{matrix}(k=1,2,\ldots,n)\\ j=k+1,\ldots,N\end{matrix} \qquad (21)$$

In other words, after calculating each column of c's and writing it down, we obtain the corresponding b-row by dividing the c-column by its leading term. The elements of the inverse are obtained according to (9), neglecting again the elements on one side of the principal diagonal. Equations (10)-(15) still hold except that in (15) we have:

$$k_{.j} = \sum_{i=1}^{j-1} k_{ji} + \sum_{i=j}^{n} k_{ij}$$

Owing to the symmetry of $\underline{M}$, equations (1) and (2) can be adjusted to exhibit this property. Let us define a diagonal matrix $\underline{Q}$

$$W = \left[w_{ij}\right] = \left[\delta_{ij} c_{ij}\right] \qquad (22)$$

which is obtained from the diagonal elements of $\underline{X}$ as defined in (18). Then:

$$\underline{Y} = \underline{W}^{-1}\underline{X}' \qquad (23)$$

Thus we obtain the following factorizations:

$$\underline{M} = \underline{X}.\underline{W}^{-1}\underline{X}' \qquad \underline{M}^{-1} = \underline{X}'^{-1}\underline{W}.\underline{X} \qquad (24)$$

Comparing the present method with the classical, it will be clear that the extra work involved here is the registration of the matrix (17), which had to be calculated in any case to obtain (16). Again we can register the b's on an additional sheet in a column-wise form, as was mentioned in the asymmetric case.

IV. ADDITION & DELETION OF ROWS AND COLUMNS

If it is required to obtain the inverse of a submatrix of $\underline{M}$ occupying the first n' ($< n$) rows and columns, the only step needed besides those involved in the previous two sections is to calculate the elements of the new inverse as:

$$k'_{ij} = \sum_{h=1}^{n-1} c_{n+i,h} b_{h,n+j} \qquad (i,j=1,\ldots,n') \qquad (25)$$

This formula is analogous to (9) and it is related to it through:

$$k_{ij} = k'_{ij} + \sum_{h=n'+1}^{n} c_{n+i,h} b_{h,n+j} \qquad (i,j=1,\ldots,n') \qquad (26)$$

This latter formula can be used to obtain the k'_{ij} from k_{ij}. If the decision on n' & n is taken before computation is started, both k'_{ij} and k_{ij} can be obtained from a single sum-product machine operation.

The only necessary condition is that the order of the columns and rows 1,2,...,n permits obtaining the required submatrix by merely deleting the columns and rows nos. n'+1,...,n. Extensions to more than one submatrix, possessing the same property are obvious. The checks (15) on the elements of the new inverses can be easily obtained on replacing n by n'. These rules apply whether the matrix $\underline{M}$ is symmetric or not.

Equations (21) serves to obtain the inverse of a bordered matrix using the inverse of the matrix itself. The extra rows and columns are written below and to the righ of $\underline{M}$ occupying positions nos. : n+1,...,n', (n' > n). This requires the calculation of the c_{ij} for j = n+1, n+2,...,n' (j = 1,...,n being already computed), then rewriting n+1,...2n as n'+1,..,n'+n and adding similar elements up to 2n'. In all cases the extra i's are i=n+1,..,n'. Similarly the extra b_{kj} are those corresponding to: k=n+1,..,n' and j=n+1,..,n & n'+n+1,..2n' as has been shown above. The elements of the inverse can then be obtained through the relations (25) and (26).

The Applications of Rules of Partitioned Matrices:

Suppose that we have a matrix $\underline{P}$ which is partitioned as follows:

$$\underline{P} = \begin{bmatrix} \underline{M} & \underline{F} \\ \underline{G} & \underline{H} \end{bmatrix} \qquad (\underline{M} \ \& \ \underline{H} \text{ square}) \qquad (27)$$

Let:

$$\underline{M}^{-1} = \underline{K}, \qquad \underline{Q} = (\underline{H} - \underline{G}.\underline{K}.\underline{F})^{-1} \qquad (28)$$

then the inverse of $\underline{P}$ can be written as follows:

$$\underline{R} = \underline{P}^{-1} = \begin{bmatrix} \underline{K} + (\underline{K}.\underline{F})\underline{Q}\,(\underline{G}.\underline{K}) & -(\underline{K}.\underline{F})\underline{Q} \\ -\underline{Q}(\underline{G}.\underline{K}) & \underline{Q} \end{bmatrix} \qquad (29)$$

Now suppose that $\underline{H}$ is a scalar, h say. Then $\underline{P}$ can be written as:

$$\underline{P} = \begin{bmatrix} \underline{M} & \underline{f}' \\ \underline{g} & h \end{bmatrix}$$

where $\underline{f}'$ is a column-vector, and $\underline{g}$ a row-vector. Since $\underline{K}$ has been already computed, we calculate the vectors:

$$\underline{r} = \underline{g}.\underline{k} \qquad \& \qquad \underline{s}' = \underline{K}.\underline{f}'$$

and the scalar:

$$t = \underline{g}.\underline{K}.\underline{f}' = \underline{r}.\underline{f}' = \underline{g}.\underline{s}'$$

The matrix $\underline{Q}$ becomes also a scalar and is found as:

$$1/q = h - t, \qquad \therefore \qquad q = \frac{1}{h - t}$$

This scalar is then to be multiplied into $\underline{r}$ & $\underline{s}'$. Further we calculate the matrix obtained by

$$\underline{s}'\,(q.\underline{r})$$

The inverse is then:

$$\underline{R} = \begin{bmatrix} \underline{K} + \underline{s}'q.\underline{r} & -\underline{s}'q \\ -q.\underline{r} & q \end{bmatrix}$$

by (29).

Thus we have used the above-mentioned rules of inverting partitioned matrices in building up the inverse when one row and one column are added. The same rules can be applied to the case where $\underline{H}$ is not a scalar but a square matrix. If $\underline{P}$ is of a high order and a number of computers can be employed, laws of partitioning can be applied according to similar steps.

The rule can be also applied to the case where $\underline{P}$ is triangular. Suppose that $\underline{F} = \underline{0}$. For $\underline{P}$ to be triangular, both $\underline{M}$ and $\underline{H}$ should be triangular also, but we need not assume that for the moment. The inverse $\underline{R}$ becomes:

$$\underline{R} = \begin{bmatrix} \underline{K} & \underline{0} \\ -\underline{H}^{-1}\underline{G}.\underline{K} & \underline{H}^{-1} \end{bmatrix} \tag{30}$$

This relation can be applied starting from the first two rows and columns, then building up the complete inverse step by step adding one row and one column each time. If the matrix $\underline{P}$ can be partitioned according to (27) with $\underline{M}$ triangular and $\underline{F} = \underline{0}$, we can apply (30), calculating $\underline{K}$ & $\underline{H}^{-1}$ by the previous methods then obtaining $\underline{R}$ from (30). These rules were used by the present author to obtain the inverse of a 33 X 33 technical matrix.

Let us now consider the <u>case of deletion</u> of rows and columns. Suppose that after the inverse $\underline{R}$ of $\underline{P}$ has been obtained we want to obtain the inverse $\underline{K}$ of a submatrix $\underline{M}$ of $\underline{P}$. (Rearranging rows and columns we can bring $\underline{M}$ to the position indicated in (27). The inverse $\underline{R}$ is now partitioned as follows:

$$\underline{R} = \begin{bmatrix} \underline{S} & \underline{T} \\ \underline{U} & \underline{Q} \end{bmatrix}$$

Comparing this partitioning with (29) it can be seen that:

$$\underline{T} = -\underline{K}.\underline{F}.\underline{Q}\ , \qquad \underline{U} = -\underline{Q}.\underline{G}.\underline{K}$$

hence,

$$\underline{S} = \underline{K} + \underline{T}.\underline{Q}^{-1}\underline{U}$$

This means that we can obtain $\underline{K}$ as follows:

$$\underline{K} = \underline{M}^{-1} = \underline{S} - \underline{T}.\underline{Q}^{-1}\underline{U} \tag{31}$$

This means that we have to invert $\underline{Q}$, and calculate the product in the second member of the R.H.S. of (31) and subtract the product from the part $\underline{S}$ of the inverse. If $\underline{Q}$ is a scalar the same rule applies noticing that $\underline{T}$ will be then a column vector and $\underline{U}$ a row vector.

V - APPLICATIONS TO LEAST-SQUARES REGRESSION

Let the (n') independent variables be denoted by z_i (i=1,...,n') and the dependent variable by y. The matrix $\underline{P}$ defined[1] above is now replaced by the matrix of the moments $(\sum_{t=1}^{T} x_{jt}x_{kt})$ where the x's are the deviations of the z's and of y from their respective means. Thus $\underline{P}$ can be partitioned as follows:

$$\underline{P} = \begin{bmatrix} \underline{M}_{zz} & \underline{m}_{zy} \\ \underline{m}_{yz} & m_{yy} \end{bmatrix} \qquad (32)$$

where y is treated as the n-th variable (n = n'+1). To the right of $\underline{P}$ we write the unit matrix of order n'n' in the first n' rows leaving the n-th row empty. Row sums and checks are computed as before, (equation 20), except for the n-th row where the unit element is replaced by a zero.

The c's are calculated in the first n columns for the following N = 2n'+1 rows as in the symetric case. The b's are calculated by (21) for the first n' rows only, i.e., leaving out the n-th row corresponding to the n-th column of the c's. The elements $c_{n+i,n}$ in this latter column give the regression coefficients

$$(c_{n+i,n}) = -\underline{M}_{zz}^{-1}\underline{m}_{zy} \qquad (i=1,\dots,n') \qquad (33)$$

The element $c_{N+1,1} = 1$ is the coefficient of the dependent variable y in the regression equation:

$$\sum c_{n+i,n} z_i + c_{N+1,n} y = u \qquad (34)$$

where u is the residual in the equation.

The residual variance is simply:

$$s^2 = c_{nn}/(T-n) \qquad (T = \text{no. of observations}) \qquad (35)$$

This can be seen from the expressions (28) and (29) for inverting partitioned matrices. For if $\underline{H}$ in (27) stands for the scalar m_{yy} then $\underline{G} = \underline{F}' = \underline{m}_{yz}$ and $\underline{Q}$ will be $(m_{yy} - \underline{m}_{yz}\underline{M}_{zz}^{-1}\underline{m}_{zy})^{-1}$. But $\underline{Q}$ will be obtained by the element $b_{n,N+1}$ (which has not been calculated):

$$b_{n,N+1} = c_{N+1,n}/c_{nn} = 1/c_{nn}$$

i.e.,

$$c_{nn} = (m_{yy} - \underline{m}_{yz}\underline{M}_{zz}^{-1}\underline{m}_{zy}) = \text{sumsquare of residuals}$$

as is well known from least-squares theory. Hence (35) follows.

In order to obtain the covariance matrix of the regression coefficients, we first compute the inverse $\underline{K} = \underline{M}_{zz}^{-1}$ by (25). We then calculate the scalar $\frac{T}{T-n} \cdot c_{nn}$. The covariance matrix is

$$\underline{V} = \left(\frac{T}{T-n} \cdot c_{nn}\right) \cdot \underline{K} \qquad (36)$$

Finally, the coefficient of multiple correlation is:

$$R = \left\{(m_{yy} - \frac{T}{T-n} \cdot c_{nn})/m_{yy}\right\}^{\frac{1}{2}} \qquad (37)$$

In some cases we are interexted in estimating the regression of a vector of m dependent variables y_i on a vecor of independent variables z_i. In such cases the vector $\underline{m}_{yz}$ is replaced by a matrix $\underline{M}_{yz}$ of order m.n'., occupying the first n' columns of the rows nos.: n'+1,n'+2,..., n'+m=n, and similarly for its transpose $\underline{M}_{zy}$. The element m_{yy} in (32) is replaced by the m.m matrix $\underline{M}_{yy}$. To right of $\underline{P}$ we again write the unit matrix of order n'n'. The elements c_{ij} for j=1,...,n' are obtained by (21). The inverse of $\underline{M}_{zz}$ is found by (25).

Defining the elements c_{ij} for i,j=n'+1,..., n as:

$$c_{ij} = a_{ij} - \sum_{k=1}^{n'} c_{ik} b_{kj} \qquad (i,j=n'+1,\ldots,n) \qquad (38)$$

we obtain the elements of the residual covariance matrix:

$$\underline{W}_{yy} = \underline{M}_{yy} - \underline{M}_{yz}\underline{M}_{zz}^{-1}\underline{M}_{zy} \qquad (39)$$

The elements c_{ij} calculated by (38) for i=n+1, n+2,...,n+n' and j = n'+1,...,n, give the elements of the regression coefficients matrix

$$-\underline{M}_{zz}^{-1}\underline{M}_{zy} \qquad (40)$$

These relations are simply straightforward extensions of the one-variable case.

Example:
Let the moment matrix be

	z_1	z_2	y
z_1	5.864665		
z_2	6.602500	8.250000	
z	4.734635	5.564500	3.983969

Suppose that we want to calculate the regression of y on the z's, knowing that T = 20. The computations are represented in Table (2):

Table (2) - Calculation of regression

	1	2	3	4	5	(&Check)
1	5.864665			1.000000	0	18.201800
2	6.602500	8.250000		0	1.000000	21.417000
3	4.734635	5.564500	3.983969	-	-	14.283104
1	5.864665	1.125810	0.807316	0.170513		3.103139
2	6.602500	0.816839	0.286716	-1.378252	1.224231	1.132695
3	4.734635	0.234201	0.094473 →	= Residual moment		
4	1.000000	-1.125810	-0.484528*	1.722163		Inverse M_{zz}^{-1}
5	0	1.000000	-0.286716*	-1.378252	1.224231	
Σ_1	1.000000	-0.125810		0.343911	-0.154021	
Σ	18.201800	0.925230	-0.676771			

* Regression Coefficients

$s^2 = 0.094473/17 = 0.005557$

Hence covariance matrix is: →

0.009570	
-0.007659	0.006803

For the multiple correlation coefficient:

$$R^2 = (3.983969 - 20 \times 0.005557)/3.983969 = 3.872829/3.983969 = 0.9721$$

VI-FURTHER APPLICATIONS: THE MULTIPLEX METHOD

For the purposes of the multiplex method of linear programming successive inverses are required to obtain regressions for sets of variables increasing by one each time. For example, Table (3) contains a part of the data included in Prof. Frisch's Memo. no. 8 (of 4/1/1958) "Data for a numerical example of multiplex method in macroeconomic linear programming". It is noticed that in this case we are interested only in the regression coefficients and not in their covariances. Hence we do not add the unit matrix as usual. The successive sets of regression coefficients are registered below the table, while the 6-th column (and row) stands for the "dependent" variable - in the regression analysis sense.