The least squares problem and pseudo-inverses

G. Peters and J. H. Wilkinson

National Physical Laboratory, Teddington, Middlesex

This paper presents a number of the most efficient mothods for computing the pseudo-inverse of an $m \times n$ matrix, developing them from a uniform standpoint. It shows that these are the natural extentions of the more common methods for inverting an $n \times n$ matrix.

(Received July 1969)

1. Introduction

In recent years a very large number of papers have been published describing algorithms for computing the pseudo-inverse of an $m \times n$ matrix. A selection from these, marked with an asterisk, is included in the references. The algorithms have generally been presented in isolation and in many cases the derivation has appeared to us to be obscure.

A somewhat similar situation existed at one time in connexion with direct methods for inverting non-singular square matrices and for solving related systems of linear equations, but as the result of the work of Householder and others the interrelationships between the various methods are now fully exposed. It is hoped that our paper will take a step in this direction for the pseudo-inverse problem.

2. The minimal least squares problem

We shall be concerned with only one vector norm, the *l*2 or Euclidean norm, defined by

$$||x|| = (\Sigma|x_i|^2)^{1/2}. \tag{2.1}$$

This is, of course, the ordinary Euclidean length. We shall use the notation A^H, x^H to denote the complex conjugate transposes of A and x respectively, so that x^H is a row vector with components $\bar{x}_i$. We note that

$$x^Hx = \Sigma\bar{x}_ix_i = \Sigma|x_i|^2 = ||x||^2, \tag{2.2}$$

and

$$\begin{aligned} ||x + y||^2 &= (x^H + y^H)(x + y) \\ &= ||x||^2 + ||y||^2 + x^Hy + y^Hx \\ &= ||x||^2 + ||y||^2 + 2\mathscr{R}(x^Hy). \end{aligned} \tag{2.3}$$

When A and x are real we use the notation A^T and x^T in place of A^H and x^H.

The pseudo-inverse of an $m \times n$ matrix A is commonly defined as the $n \times m$ matrix X such that

$$XAX = X, \quad AXA = A, \quad \text{and } AX \text{ and } XA \text{ are Hermitian.} \tag{2.4}$$

Clearly when A is non-singular, the matrix $X = A^{-1}$ satisfies these relations. This definition has always appeared to us to be somewhat artificial and we prefer to introduce the pseudo-inverse via the least squares problem which may be stated explicitly in the following terms:

'Given an $m \times n$ matrix A and a vector b of order m, how shall we determine a vector x of order n such that the residual vector $b - Ax$ has the minimum norm?' In general the vector x is not unique and it is pertinent to ask the further question: 'Which of the vectors x giving the minimum residual has the minimum norm?' We shall refer to this vector as the *minimal least squares solution*. It turns out that this vector is unique and is given by Xb where X is the pseudo-inverse. 'Pseudo-inverse' is therefore quite a natural name, since when A is square and non-singular the solution is clearly $A^{-1}b$.

Theorem 1:

A necessary and sufficient condition that $||b - Ax||$ shall be a minimum is that

$$A^H(b - Ax) = 0. \tag{2.5}$$

Sufficient:

Suppose $A^H(b - Ax) = 0$. The length of the residual $r(\delta)$ corresponding to any other vector $x + \delta$ is given by

$$\begin{aligned} ||r(\delta)||^2 &= ||b - Ax - A\delta||^2 \\ &= ||b - Ax||^2 + ||A\delta||^2 - 2\mathscr{R}((b - Ax)^HA\delta) \\ &= ||b - Ax||^2 + ||A\delta||^2. \end{aligned} \tag{2.6}$$

Since $||A\delta||$ is non-negative, the residual $b - Ax$ is a minimum. Unless $A\delta = 0$ the residual corresponding to $x + \delta$ is actually greater than that corresponding to x; the only other vectors giving a minimum residual are therefore of the form $x + \delta$ with $A\delta = 0$. Hence if the columns of A are independent there cannot be more than one vector giving the minimum residual.

Necessary:

Suppose x gives a minimum residual but

$$A^H(b - Ax) = z \neq 0. \tag{2.7}$$

Consider the residual corresponding to $x + \epsilon z$; we have

$$\begin{aligned} ||b - Ax - A\epsilon z|| &= ||b - Ax||^2 - \epsilon z^HA^H(b - Ax) \\ &\quad - \epsilon(b - Ax)^HAz + \epsilon^2||Az||^2 \\ &= ||b - Ax||^2 - 2\epsilon||z||^2 + \epsilon^2||Az||^2 \end{aligned} \tag{2.8}$$

For sufficiently small positive ϵ the right-hand side is clearly less than $||b - Ax||^2$, contradicting the hypothesis.

Corollary:

When A^HA is non-singular the least squares solution is

$$x = (A^HA)^{-1}A^Hb \tag{2.9}$$

conventionally given by the normal equations.

3. Explicit expression for the pseudo-inverse

If the $m \times n$ matrix is of rank r then it can be factorised in the form

$$A = BC \tag{3.1}$$

where B is an $m \times r$ matrix and C an $r \times n$ matrix and both are of rank r. We shall not prove this formally since the algorithms we describe give explicit factorisations of this kind. The matrices B^HB and CC^H are both Hermitian positive definite $r \times r$ matrices, and hence non-singular. For if B^HB is singular, there exists an x such that $B^HBx = 0$, giving $x^HB^HBx = 0$. But this implies that $Bx = 0$, which is impossible since B is of rank r.

The factorisation is not unique since if Y is any non-singular $r \times r$ matrix, BY^{-1} and YC are also factors of the specified type. In fact all other factorisations are derived in this way. For if

$$A = BC = B_1C_1 \tag{3.2}$$

we have

$$B^HBC = B^HB_1C_1 \quad \text{giving} \quad C = (B^HB)^{-1}B^HB_1C_1 = ZC_1 \quad \text{(say)}. \tag{3.3}$$

Z is an $r \times r$ matrix and it must be non-singular, because if it were of rank less than r the same would be true of C. Hence we have

$$BZC_1 = B_1C_1 \quad \text{giving} \quad BZC_1C_1^H = B_1C_1C_1^H, \tag{3.4}$$

from which it follows that $B_1 = BZ$ since $C_1C_1^H$ is non-singular.

The minimal least squares solution can be given explicitly in terms of any of the BC factorisations of A, as we now show. Suppose x is any vector giving a minimal residual. Then $A^H(b - Ax) = 0$ giving

$$A^HAx = A^Hb \quad \text{or} \quad C^HB^HBCx = C^HB^Hb. \tag{3.5}$$

From this we have

$$CC^HB^HBCx = CC^HB^Hb \quad \text{giving} \quad Cx = (B^HB)^{-1}B^Hb \tag{3.6}$$

from the non-singularity of B^HB and CC^H. Now one solution of the equation $Cx = v$ is obviously given by $x = C^H(CC^H)^{-1}v$, and hence from (3.6) a solution of the least squares problem is $x = C^H(CC^H)^{-1}(B^HB)^{-1}B^Hb$, and all other solutions are given by $x + \delta$, where $C\delta = 0$.

We now show that this x is indeed the *minimal* least squares solution. For we have

$$||x + \delta||^2 = ||x||^2 + ||\delta||^2 + 2\mathscr{R}(x^H\delta) \tag{3.7}$$

and

$$x^H\delta = b^HB(B^HB)^{-1}(CC^H)^{-1}C\delta = 0 \tag{3.8}$$

since $C\delta$ must be null if $x + \delta$ is a least squares solution. This shows that δ must be zero for a minimal least squares solution and

$$x = C^H(CC^H)^{-1}(B^HB)^{-1}B^Hb \tag{3.9}$$

is therefore the *unique* minimal least squares solution.

The matrix

$$X = C^H(CC^H)^{-1}(B^HB)^{-1}B^H \tag{3.10}$$

is the pseudo-inverse of A. Its very derivation shows that it must be independent of the particular BC factorisation chosen and this can be verified by replacing B by BY^{-1} and C by YC, where Y is any non-singular $r \times r$ matrix. It may also be verified that X satisfies the conditions (2.4).

4. Formal algorithms

It is well known that the problem of determining the rank of a matrix is far from trivial when rounding errors are involved, as invariably they are on a digital computer. For convenience, however, we concentrate on the purely formal aspects of the problem until Section 9; indeed most of the published algorithms do just this and ignore the practical difficulties of determining the rank.

Several of the best algorithms for inverting a non-singular $n \times n$ matrix or solving a system of equations are based on factorisations $A = BC$ of A, where B and C are non-singular and are easily invertible—for example, upper and lower triangular matrices and unitary (orthogonal) matrices. (In these problems the generalised inverse given in (3.9) reduces to $C^{-1}B^{-1}$.) Corresponding to each of the well-known methods for inverting matrices based on such factorisations we might expect to find an analogous method for computing a pseudo-inverse, and we shall show that this expectation is justified.

When solving linear equations it is more economical to work directly with the factors B and C by solving

$$By = b, \quad Cx = y \tag{4.1}$$

rather than computing $C^{-1}B^{-1}$ explicitly. Similarly, if we are directly concerned with the least squares problem it is uneconomical to compute the pseudo-inverse explicitly. We observe that we may write

$$C^H(CC^H)^{-1}(B^HB)^{-1}B^H = C^H(B^HBCC^H)^{-1}B^H. \tag{4.2}$$

Hence we may compute x given by (3.9) in the steps

$$u = B^Hb, \quad (B^HBCC^H)v = u, \quad x = C^Hv, \tag{4.3}$$

and to solve $(B^HBCC^H)v = u$ we require only some factorisation of the matrix pre-multiplying v, not its explicit inverse. We notice that

$$B^HBCC^H = B^HAC^H. \tag{4.4}$$

When solving systems of equations it is quite common to employ some form of pivoting in order to achieve greater numerical stability. This has the effect that we determine a factorisation of a matrix $\tilde{A}$, rather than of A itself, where $\tilde{A}$ is derived from A by suitably permuting its rows and/or columns. What we therefore achieve is a factorisation

$$\tilde{A} = P_1AP_2 = BC \quad \text{or} \quad A = P_1^TBCP_2^T, \tag{4.5}$$

where P_1 and P_2 are permutation matrices. Clearly

$$\tilde{A}^{-1} = (P_1AP_2)^{-1} = P_2^{-1}A^{-1}P_1^{-1} = P_2^TA^{-1}P_1^T, \tag{4.6}$$

and hence the inverse derived via B and C merely gives us the required matrix with its columns and rows permuted. The same is true of the pseudo-inverse as can be seen immediately from the explicit form (3.10). We may therefore use row and column interchanges freely in our factorisations.*

* It is evident that if X is the pseudo-inverse of A, X^H is that of A^H.

5. The LU and related factorisations

The simplest of the factorisations used in solving linear systems are those related to Gaussian elimination. These effectively give a decomposition of $\tilde{A}$ of the form $\tilde{A} = LU$ where L is lower-triangular and U is upper-triangular. We may choose either L or U to have a unit diagonal or alternatively we may write $\tilde{A} = LDU$ where D is a diagonal matrix and both L and U have unit diagonals.

Corresponding techniques may be used for the pseudo-inverse but since we have to determine the rank, the pivotal strategy is particularly important. We therefore describe one of these algorithms explicitly. It consists of r major steps in which $A = A_1$ is reduced successively to $A_2, A_3, \ldots, A_{r+1}$. Before the sth major step we have a matrix A_s of the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ 0 & 0 & a_{33} & a_{34} & a_{35} & a_{36} \\ 0 & 0 & a_{43} & a_{44} & a_{45} & a_{46} \end{bmatrix} \tag{5.1}$$

when $m = 4$, $n = 6$, $s = 3$. Each a_{ij} should have an upper suffix, s, but for convenience we omit this. In practice quantities m_{ij} derived in the previous steps will be stored in the positions occupied by the zeros, so that the stored array is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ m_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ m_{31} & m_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ m_{41} & m_{42} & a_{43} & a_{44} & a_{45} & a_{46} \end{bmatrix}. \tag{5.2}$$

The sth step is then as follows.

Determine the element of maximum modulus among the current a_{ij} with $i, j \geqslant s$. Suppose this element is $a_{s',s''}$. (If there is more than one, we take that with smallest s' and s''.) If this maximum element is zero the reduction is complete and $s = r + 1$. Otherwise interchange rows s and s' and columns s and s'' *in the complete* $m \times n$ *rectangular array*, and for each value of i from $s + 1$ to m do the following:

(i) Compute $m_{is} = a_{is}/a_{ss}$ and store it in position (i, s). Note $|m_{is}| \leqslant 1$.

(ii) For each value of j from $s + 1$ to n compute a new a_{ij} given by $a_{ij} - m_{is}a_{sj}$ and overwrite it on the old a_{ij}.

It is easy to see that *with exact computation* this process terminates with A_{r+1}; when $m = 6, n = 4, r = 2$ the final stored array has the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ m_{21} & a_{22} & a_{23} & a_{24} \\ m_{31} & m_{32} & 0 & 0 \\ m_{41} & m_{42} & 0 & 0 \\ m_{51} & m_{52} & 0 & 0 \\ m_{61} & m_{62} & 0 & 0 \end{bmatrix}. \tag{5.3}$$

Notice that the m_{ij} will also have been affected by the row interchanges but m_{ij} always denotes the element currently stored in position (i, j). Exactly as in Gaussian elimination one can now show that $\tilde{A}$ (i.e. the permuted A) has been factorised in the form

$$\tilde{A} = L_{r+1}A_{r+1} \tag{5.4}$$

where for (5.3) L_{r+1} and A_{r+1} have the forms

$$\left.\begin{aligned} L_{r+1} &= \begin{bmatrix} 1 & & & & & \\ m_{21} & 1 & & & & \\ m_{31} & m_{32} & 1 & & & \\ m_{41} & m_{42} & 0 & 1 & & \\ m_{51} & m_{52} & 0 & 0 & 1 & \\ m_{61} & m_{62} & 0 & 0 & 0 & 1 \end{bmatrix} \\ A_{r+1} &= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}\right\} \tag{5.5}$$

Clearly columns $r + 1$ to m of L_{r+1} and rows $r + 1$ to m of A_{r+1} play no part in the product $L_{r+1}A_{r+1}$ and we may write

$$\tilde{A} = LU, \tag{5.6}$$

where L consists of the first r columns of L_{r+1} and U consists of the first r rows of A_{r+1}. Thus L is an $m \times r$ unit lower-trapezoidal matrix, U is an $r \times n$ upper-trapezoidal matrix, both are of rank r and we have a factorisation of the form (3.1). The corresponding minimal least squares solution (3.9) is

$$x = U^H(UU^H)^{-1}(L^HL)^{-1}L^Hb \tag{5.7}$$

$$= U^H(L^HLUU^H)^{-1}L^Hb. \tag{5.8}$$

The matrix $Y = L^HLUU^H$ is of dimension $r \times r$; we do not need its explicit inverse, but merely an appropriate factorisation whereby to compute the right-hand side of (5.8). (In the present context an LU factorisation of Y would give the whole procedure a pleasing unity!) L^HL and UU^H are Hermitian positive definite and in the real case will be real and symmetric positive definite; the Cholesky decompositions of UU^H and L^HL could therefore be used in conjunction with (5.7).

Obviously we may write

$$U = D\hat{U}, \tag{5.9}$$

where $D = \text{diag}(u_{ii})$. The matrix $\hat{U}$ is then an $r \times n$ unit upper-trapezoidal matrix and all its elements satisfy the relation $|\hat{u}_{ij}| \leqslant 1$ because of the pivotal strategy. With the $LD\hat{U}$ decomposition equations (5.7) and (5.8) reduce to

$$x = \hat{U}^H(D\hat{U}\hat{U}^H)^{-1}(L^HL)^{-1}L^Hb \tag{5.10}$$

$$= \hat{U}^H(L^HLD\hat{U}\hat{U}^H)^{-1}L^Hb, \tag{5.11}$$

since a factor D^H cancels in the reduction. It turns out that there are advantages as regards numerical stability in using (5.10) and (5.11) rather than (5.7) and (5.8) and we discuss these later.

There is a similar algorithm based on the Gauss–Jordan factorisation but since this has been described by Noble (1966) in terms which are reasonably closely related to those used here, we refer the reader to Noble's

paper. This approach also has advantages of the type associated with the $LD\hat{U}$ decomposition.

6. Factorisations involving unitary matrices

Householder (1958) and Givens (1958) have described algorithms for solving systems of equations which depend on unitary transformations with matrices of the type $I - 2ww^H$ and plane rotations respectively. They are slightly better than elimination methods as regards numerical stability, but Householder's method requires twice as much work and Givens' method four times as much. There are analogous methods for the minimal least squares problem but the balance of work is somewhat different. We discuss only the Householder procedure; from this the corresponding Givens algorithm is obvious.

The Householder reduction consists of r major steps in which $A = A_1$ is reduced successively to $A_2, A_3, \ldots, A_{r+1}$. At the beginning of the sth step A_1 has been reduced to A_s of the form illustrated in (5.1). The sth major step is as follows:

Compute $\sigma_i = \sum_{j=s}^{m} |a_{ji}|^2$ for each value of i from s to n.

Let $\sigma_{s'}$ be the maximum of these sums. (If there are several equal to the maximum we take the first.) If this maximum is zero, $s = r + 1$ and the reduction is complete. Otherwise columns s' and s are interchanged in the full array, and the resulting A_s is premultiplied by P_s to give A_{s+1}, where P_s is of the form

$$P_s = I - 2w_s w_s^T, \quad ||w_s|| = 1. \tag{6.1}$$

Here w_s is chosen to annihilate the elements a_{is} $(i = s + 1, \ldots, m)$ of A_s as described, for example, by Wilkinson (1965); it has its first $s - 1$ components equal to zero, and the pre-multiplication leaves rows 1 to $s - 1$ and the zeros of A_s unaltered. In practice the non-zero elements of w_s apart from its sth element may be stored in the positions occupied by the zeros which pre-multiplication introduces into A_{s+1}. Hence we have finally

$$P_r \ldots P_2 P_1 \tilde{A} = A_{r+1} \tag{6.2}$$

where, when $m = 5$, $n = 4$, $r = 2$, A_{r+1} has the form

$$\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.3}$$

and $\tilde{A}$ is A with its columns permuted. We may write

$$\tilde{A} = P_1 P_2 \ldots P_r A_{r+1} = Q A_{r+1}, \tag{6.4}$$

where

$$Q = P_1 P_2 \ldots P_r; \tag{6.5}$$

clearly Q is a unitary matrix. In the product QA_{r+1} columns $r + 1$ to n of Q and rows $r + 1$ to m of A_{r+1} play no part and we may write

$$\tilde{A} = (QI_{m,r})U, \tag{6.6}$$

where $I_{m,r}$ consists of the first r columns of the identity matrix I_m and U is the $r \times n$ upper-trapezoidal matrix given by the first r rows of A_{r+1}. The minimal least squares solution (3.9) is then given by

$$\begin{aligned} x &= U^H(UU^H)^{-1}(I_{m,r}^T Q^H Q I_{m,r})^{-1} I_{m,r}^T Q^H b \\ &= U^H(UU^H)^{-1} I_{m,r}^T Q^H b \end{aligned} \tag{6.7}$$

since $I_{m,r}^T Q^H Q I_{m,r}$ is I_r. Comparison with (5.7) shows that we are compensated for the extra work involved in the factorisation by the simplification in the expression for the pseudo-inverse.

The matrix $I_{m,r}^T Q^H$ is not derived explicitly when computing x from (6.7). Since $Q^H = P_r P_{r-1} \ldots P_1$ we merely pre-multiply b successively by $P_1, P_2, \ldots, P_r$ and then take the first r components. When the pseudo-inverse is required explicitly $I_{m,r}^T$ is post-multiplied successively by $P_r, \ldots, P_2, P_1$. In neither case is $(UU^H)^{-1}$ computed explicitly. We merely need to factorise UU^H and since this is a positive definite Hermitian matrix the Cholesky factorisation is the most appropriate.

To improve numerical stability U may be factorised in the form $D\hat{U}$ where D is an $r \times r$ diagonal matrix and $\hat{U}$ is a unit upper-trapezoidal $r \times n$ matrix. Because of the pivoting $|\hat{u}_{ij}| \leqslant 1$ for all relevant i, j. The factor $U^H(UU^H)^{-1}$ then becomes $\hat{U}(\hat{U}\hat{U}^H)^{-1}D^{-1}$ as in (5.10).

The reduction can be extended by factorising U in the form TQ^* where T is an $r \times r$ upper triangular matrix and Q^* is an $r \times n$ matrix consisting of the first r rows of an $n \times n$ unitary matrix. This can be achieved by post-multiplying U successively by elementary Hermitian matrices $P_r^*, P_{r-1}^*, \ldots, P_1^*$. (Note that the asterisk is used merely to distinguish P_i^* from P_i used earlier.) Immediately before post-multiplying by P_s^*, the current matrix has the form

$$\begin{bmatrix} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.8}$$

when $r = 4, n = 8, s = 2$. The w_s vector associated with P_s^* is chosen to annihilate the elements $u_{s,r+1}, \ldots, u_{s,n}$, and the disposition of the zero and non-zero elements in w_s is

$$\underbrace{0\ 0\ \ldots\ 0}_{s-1}\ \ x\ \ \underbrace{0\ 0\ \ldots\ 0}_{r-s}\ \ \underbrace{x\ x\ x\ x}_{n-r} \tag{6.9}$$

After post-multiplication by P_1^* we have

$$UP_r^* \ldots P_2^* P_1^* = [T \mid O] \tag{6.10}$$

giving

$$U = [T \mid O] P_1^* P_2^* \ldots P_r^* \tag{6.11}$$

or

$$U = T I_{n,r}^T \tilde{Q} \tag{6.12}$$

where $\tilde{Q}$ is the $n \times n$ unitary matrix defined by

$$\tilde{Q} = P_1^* P_2^* \ldots P_r^*. \tag{6.13}$$

Equations (6.7) becomes

$$\begin{aligned} x &= \tilde{Q}^H I_{n,r} T^H (T I_{n,r}^T \tilde{Q} \tilde{Q}^H I_{n,r} T^H)^{-1} (I_{m,r}^T Q^H Q I_{m,r}) I_{m,r}^T Q^H b \\ &= \tilde{Q}^H I_{n,r} T^{-1} I_{m,r}^T Q^H b \end{aligned} \tag{6.14}$$

in which both Q and $\tilde{Q}$ are used in their factorised forms (6.5) and (6.13).

7. The Gram–Schmidt factorisation

A factorisation similar to that given by the Householder method is provided by the Gram–Schmidt orthogonalisation process. In this method we explicitly determine a set of r orthogonal vectors $q_1, q_2, \ldots, q_r$ which span the same space as the n columns $a_1 a_2 \ldots a_n$. Again the process consists of r major steps in which the matrix $A = A_1$ is transformed successively to $A_2, A_3, \ldots, A_{r+1}$. At the beginning of the sth step A_s has columns denoted by

$$(q_1 q_2 \ldots q_{s-1} a_s^{(s)} a_{s+1}^{(s)} \ldots a_n^{(s)}) \tag{7.1}$$

where $q_1, \ldots, q_{s-1}$ are orthonormal vectors and $a_s^{(s)}, \ldots, a_n^{(s)}$ are modified versions of the corresponding original columns $a_s^{(1)}, \ldots, a_n^{(1)}$ of A_1. The sth step is then as follows:

Compute $||a_i^{(s)}||$ for $i = s, \ldots, n$. Let the maximum of these be $||a_{s'}^{(s)}||$. (If there is more than one, $||a_{s'}^{(s)}||$ is the first of these.) If this maximum is zero all columns $a_s^{(s)}, \ldots, a_n^{(s)}$ are null, the reduction is complete and $s = r + 1$. Otherwise interchange columns s and s'. (We shall still call the ith column $a_i^{(s)}$ after this interchange.) Now replace the current $a_s^{(s)}$ by the unit vector $q_s = a_s^{(s)}/||a_s^{(s)}||$. Define u_{ss} to be $||a_s^{(s)}||$. For each value of i from $s + 1$ to n compute

$$a_i^{(s+1)} = a_i^{(s)} - u_{si} q_s, \quad \text{where} \quad u_{si} = q_s^H a_i^{(s)} \tag{7.2}$$

i.e. the current $a_i^{(s)}$ is orthogonalised with respect to q_s.

The process terminates with the matrix A_{r+1} which is of the form

$$(q_1 q_2 \ldots q_r 0 \ldots 0) \tag{7.3}$$

Clearly we again have a decomposition QU of $\tilde{A}$ (which is A with its columns permuted) where Q consists of the first r columns of a unitary matrix and U is an $r \times n$ upper-trapezoidal matrix. When $r = 3$ and $n = 5$ it is

$$\tilde{A} = QU = (q_1 q_2 q_3) \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ & u_{22} & u_{23} & u_{24} & u_{25} \\ & & u_{33} & u_{34} & u_{35} \end{bmatrix}. \tag{7.4}$$

This is of the same form as the Householder decomposition (6.6), and the minimal least squares solution (3.9) is

$$x = U^H (UU^H)^{-1} (Q^H Q)^{-1} Q^H b \tag{7.5}$$

$$= U^H (UU^H)^{-1} Q^H b. \tag{7.6}$$

Here we obtain *explicitly* the matrix Q as part of a unitary matrix, whereas the product $P_1 P_2 \ldots P_r$ of (6.5) gives a full unitary matrix of which we are interested only in the first r columns. As with the LU and Householder algorithms we may write $U = D\hat{U}$.

The process described here is usually called the *modified* Gram–Schmidt to distinguish it from the classical procedure. Ignoring the column interchanges, in the classical Gram–Schmidt one has

$$a_i^{(s+1)} = a_i^{(s)} - \bar{u}_{si} q_s \quad \text{where} \quad \bar{u}_{si} = q_s^H a_i^{(1)}, \tag{7.7}$$

in place of (7.2), i.e. one uses the original $a_i^{(1)}$ at each stage to determine $\bar{u}_{si}$. Now

$$a_i^{(s)} = a_i^{(1)} - u_{1i} q_1 - u_{2i} q_2 - \ldots - u_{s-1,i} q_{s-1}$$

and hence *with exact computation*

$$u_{si} = q_s^H a_i^{(s)} = q^H a^{(1)} = \bar{u}_{si} \tag{7.8}$$

from the orthogonality of the q_j. However, the classical Gram–Schmidt process is very unstable and the modified version is much to be preferred—in fact it had almost always been used in practice (because it is more convenient), long before its superior numerical stability was appreciated. Evidence is accumulating that the modified Gram–Schmidt gives better results than Householder in spite of the fact that the latter guarantees almost exact orthogonality of the columns of Q while this is by no means true of the modified Gram–Schmidt procedure when A has ill-conditioned columns. The reasons for this phenomenon appear not to have been elucidated yet.

The vector $Q^H b$ on the right-hand side of (7.6) has the components

$$(q_1^H b, q_2^H b, \ldots, q_r^H b) = (c_1, c_2, \ldots, c_r) \quad \text{(say)}. \tag{7.9}$$

It is in the spirit of modified Gram–Schmidt to compute $c_1, \ldots, c_r$ by orthogonalising b successively using the relations

$$b = b_1, \quad c_s = q_s^H b_s, \quad b_{s+1} = b_s - c_s q_s \tag{7.10}$$

and in practice this gives better results.

8. Matrices of maximum rank

The most common least squares problem is that when $r = n < m$ so that we have an overdetermined system, but A is of maximum rank. In the decomposition $\tilde{A} = LU$ given in (5.6), L is an $m \times n$ unit lower-trapezoidal matrix and U is now a non-singular $n \times n$ *upper-triangular* matrix. (5.7) then reduces to

$$x = U^{-1} (L^H L)^{-1} L^H b. \tag{8.1}$$

It is well known that the solution $(A^H A)^{-1} A^H b$ given in (2.9) is unsatisfactory because the condition number of $A^H A$ is the square of that of A. To avoid this difficulty a QU decomposition of A is usually recommended (Golub, 1965). However, (8.1) shows that an LU factorisation also can be used beneficially: pivoting usually causes L to be well-conditioned and any ill-condition in A is wholly reflected in U. However, $U^H (UU^H)^{-1}$ reduces to U^{-1}, and we avoid squaring the condition number. The factors $(L^H L)^{-1} L^H$ do not simplify, though in the $n \times n$ linear equation case, L is a non-singular lower-triangular matrix and $(L^H L)^{-1} L^H$ simplifies to L^{-1}.

Turning now to the QU decomposition we have

$$A = QU \tag{8.2}$$

where Q consists of the first n columns of an $m \times m$ unitary matrix and U is a non-singular $n \times n$ *upper-triangular* matrix. (3.9) then gives

$$x = U^H (UU^H)^{-1} (Q^H Q)^{-1} Q^H b$$

$$= U^{-1} Q^H b. \tag{8.3}$$

Comparing (8.1) with (8.3) we see that we are compensated for the extra work involved in the QU decomposition by the simplification of the expression for the pseudo-inverse. In the non-singular $n \times n$ case, however, the factor $(L^H L)^{-1} L^H$ does simplify to L^{-1} and hence there is no compensatory gain for the QU factorisation.

The case when $r = m < n$, that is, an under-determined system of maximum rank, leads to similar results. We now have

$$A = LU \tag{8.4}$$

where L is a non-singular $m \times m$ lower-triangular matrix and U is an $m \times n$ upper-trapezoidal matrix. (5.7) then becomes

$$x = U^H(UU^H)^{-1}L^{-1}b. \tag{8.5}$$

It is more convenient now to let U have the unit diagonal elements so that L wholly reflects any ill-condition in A; nonetheless the simplification of $(L^HL)^{-1}L^H$ to L^{-1} avoids the squaring of the condition number.

If we use a unitary factorisation it is better to derive a decomposition of the type

$$A^H = QU \tag{8.6}$$

where Q consists of the first m columns of an $n \times n$ unitary matrix and U is a non-singular $m \times m$ *upper-triangular* matrix. Hence

$$A = U^HQ^H = L\tilde{Q}, \tag{8.7}$$

where L is a non-singular $m \times m$ lower-triangular matrix and $\tilde{Q}$ consists of the first m rows of an $n \times n$ unitary matrix. (3.9) now gives

$$\begin{aligned} x &= \tilde{Q}^H(\tilde{Q}Q^H)^{-1}(L^HL)^{-1}L^Hb \\ &= \tilde{Q}^HL^{-1}b \end{aligned} \tag{8.8}$$

thus more simplification takes place than with the LU decomposition thereby compensating for the extra work involved.*

We have shown that in the cases of full rank we avoid the squaring of the condition number because of simplifications that occur. These simplifications do not occur *naturally* when $r < \min(m, n)$. However, by use of factorisations of the type $U = D\hat{U}$ described in Sections 5 and 6 we *are* able to avoid this squaring of the condition number. In fact the ill-condition of A will *usually* be fully represented in U and this in turn will usually be fully represented in D. In the decomposition of Section 5 both L and $\hat{U}$ will *usually* be well-conditioned as a result of the pivotal strategy. Because of the cancellation of factors D^H and $(D^H)^{-1}$ in the explicit expression for the pseudo-inverse, worsening of the condition is usually avoided. With the factorisation described in (6.14) the squaring of the condition number is certainly avoided by the cancellation of the factors T^H and $(T^H)^{-1}$.

9. Practical considerations

(i) So far we have deliberately avoided the most difficult practical problem, the determination of the rank. If the original matrix is exact and we use exact arithmetic no such problem arises. If the original matrix is not exact and/or rounding errors are involved in the factorisations we have to decide when the 'remaining' elements can be regarded as zero during the course of the reduction. In fact we have to decide precisely what is meant by the 'rank' of an inexact matrix. If we regard each individual element as having an error (or an uncertainty) of modulus bounded by ϵ, then it is reasonable to define the 'rank' of A as the minimum rank (in the precise mathematical sense) of all matrices $\tilde{A}$ defined by

$$|\tilde{a}_{ij} - a_{ij}| \leqslant \epsilon \quad (i \leqslant m, j \leqslant n). \tag{9.1}$$

One might expect that the pivotal strategies employed will ensure that after r reductions the 'remaining' elements will be small and that we need only check that they are smaller than some modest multiple of ϵ. Unfortunately it is not difficult to construct matrices for which this is not true. Consider the matrix A of order n of the type illustrated by

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ & 1 & -1 & -1 \\ & & 1 & -1 \\ & & & 1 \end{bmatrix}. \tag{9.2}$$

It is easy to show that if ϵ is added to each subdiagonal element in the first column, the determinant is $1 + (2^{n-1} - 1)\epsilon$, so that the modified matrix is singular if $\epsilon = -1/(2^{n-1} - 1)$. Hence for this value of ϵ, the 'rank' of A defined above is certainly less than n. Yet if we perform Gaussian elimination with pivoting as we have described it, $L = I$ and $U = A$. There are no small elements in U and we receive no indication that the 'rank' is less than n. However, the algorithms described in Sections 6 and 7 show immediately that when n is not small A is close to a singular matrix.

Although one of the writers was the first to introduce this example to illustrate the insidious nature of ill-conditioning, we consider that there is a tendency to exaggerate its importance; in our experience procedures based on the algorithms we have described, together with 'reasonable' criteria for recognising zero elements in the remaining matrix, have been quite satisfactory. Of course, it is idle to deny that the recognition of rank presents severe difficulties when elements of A vary widely in their orders of magnitude, but the main purpose of this paper is to present the formal aspects of the problem in a simple and unified manner and merely to indicate the practical difficulties. It should be recognised that these difficulties are largely subjective and sound decisions can only be made by somebody who appreciates the underlying physical problem.

Golub and Kahan (1964) have described an algorithm which is somewhat more effective for determining the 'rank' of an inexact matrix. The matrix A is reduced by orthogonal transformations to the form F illustrated by

$$F = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{9.3}$$

i.e. F has non-zero elements only on the diagonal and super-diagonal. F is given by

$$F = \ldots P_3P_2P_1AR_2R_3R_4 \ldots \tag{9.4}$$

where the pre-multiplications and post-multiplications are performed alternately, and introduce the zeros successively in column 1, row 1, column 2, row 2, . . . The pre-multiplying P_r are $m \times m$ matrices of the form

* M. J. D. Powell (1969) has recently given a very efficient algorithm for computing $\tilde{Q}^HL^{-1}$ explicitly when the inverse is required.

$I - 2w_r w_r^H$ and the post-multiplying R_r are $n \times n$ matrices of this type, the w_r being determined in the usual way. The singular values of A are those of F, and the latter may be found via the eigenvalues of a real symmetric tridiagonal matrix. (This technique avoids the loss of accuracy inherent in finding the singular values via the eigenvalues of AA^H.) Although this is perhaps a little more reliable than the algorithms we have described (it shows immediately that the matrix of type (9.2) has a small singular value and hence is close to a singular matrix), it is by no means certain that the extra complexity is justified. However, the singular value decomposition is of great value in solving a number of related problems (see Golub and Reinsch (1970)) and Golub's algorithm must be regarded as one of the most important in this field.

(ii) The requirements that the residual vector should be a minimum while x itself should be minimal often conflict to some extent. This has been well illustrated by the procedures produced on KDF9 embodying the algorithms described in Sections 5 to 7. Perhaps the point can be made most simply by means of a trivial example. Consider the system

$$A = \begin{bmatrix} 6 & 3{\cdot}0000\ 00000 \\ 4 & 1{\cdot}9999\ 99998 \\ 2 & 1{\cdot}0000\ 00003 \end{bmatrix} \qquad b = \begin{bmatrix} 3{\cdot}0000\ 00000 \\ 2{\cdot}0004\ 00000 \\ 0{\cdot}9994\ 00000 \end{bmatrix}$$

It is immediately evident that the vector $x_1 = 0{\cdot}4$, $x_2 = 0{\cdot}2$ gives a residual with the components

$$0, \quad 4(10^{-4}) + 4(10^{-10}), \quad -6(10^{-4}) - 6(10^{-10}),$$

while the vector $x_1 = 10^5$, $x_2 = -2(10^5) + 1$ gives a residual with components

$$0, \quad 2(10^{-9}), \quad -3(10^{-9}).$$

The second vector gives much the smaller residual but is itself very much larger. In many physical problems 'large' solution vectors are quite unacceptable. With our trivial example a solution vector very close to the first is obtained if we use a tolerance of 10^{-8} (say) in making the decisions about rank. The 'rank' will then be diagnosed as one. If tolerance of 10^{-10} (say) is adopted the 'rank' will be given as two, and a solution vector rather like the second one will be obtained.

In practice the case against the larger solution vector is a good deal stronger than we have indicated. For simplicity the elements of the 'large' solution have been taken to be exact integers. Normally this will not be true and we will derive numbers which must be rounded. On a ten-digit floating-point decimal computer the rounding errors made in numbers of the order 10^5 may be as large as $\frac{1}{2}.10^{-5}$. These rounding errors alone may give contributions to the residual of the order of 10^{-5}. Hence the potential reduction in the residual will not usually be realised in practice.

10. Numerical examples

As an illustration of the use of the procedures, experiments were carried out with the leading 7×6 matrix of a scaled version of the Hilbert matrix given in Fig. 1. (In fact 10^5 times the above matrix was used to avoid input rounding errors.) The minimal least squares solution was determined by the algorithm given by (6.8)–(6.14), corresponding to two compatible right-hand sides

$$b_1 = \begin{matrix} 8{\cdot}82882 & 5{\cdot}74002 & 4{\cdot}38867 & 3{\cdot}58787 \\ 3{\cdot}04733 & 2{\cdot}65421 & 2{\cdot}35391 \end{matrix}$$

$$b_2 = \begin{matrix} 2{\cdot}22222 & 0{\cdot}86658 & 0{\cdot}48477 & 0{\cdot}31603 \\ 0{\cdot}22451 & 0{\cdot}16861 & 0{\cdot}13169 \end{matrix}$$

the solutions being 1, 1, 1, 1, 1, 1 and 1, −1, 1, −1, 1, −1 respectively. Using a tolerance of 10^{-7} the rank was determined as 6 (which is correct if the matrix is regarded as exact) and the computed solutions and residuals were

1st solution	*residual*
$+1{\cdot}0000\ 0000\ 700_{10} + 0$	$-6{\cdot}1988\ 8305\ 664_{10} - 11$
$+9{\cdot}9999\ 9818\ 023_{10} - 1$	$+1{\cdot}1920\ 9289\ 551_{10} - 12$
$+1{\cdot}0000\ 0116\ 733_{10} + 0$	$-4{\cdot}7683\ 7158\ 203_{10} - 12$
$+9{\cdot}9999\ 7072\ 897_{10} - 1$	$-7{\cdot}1525\ 5737\ 305_{10} - 12$
$+1{\cdot}0000\ 0314\ 428_{10} + 0$	$-2{\cdot}3841\ 8579\ 102_{10} - 12$
$+9{\cdot}9999\ 8786\ 943_{10} - 1$	$+0{\cdot}0000\ 0000\ 000$
	$+2{\cdot}3841\ 8579\ 102_{10} - 12$

2nd solution	*residual*
$+9{\cdot}9999\ 9999\ 116_{10} - 1$	$+7{\cdot}1525\ 5737\ 305_{10} - 12$
$-9{\cdot}9999\ 9976\ 601_{10} - 1$	$+7{\cdot}1525\ 5737\ 305_{10} - 12$
$+9{\cdot}9999\ 9849\ 131_{10} - 1$	$+3{\cdot}5762\ 7868\ 652_{10} - 12$
$-9{\cdot}9999\ 9620\ 670_{10} - 1$	$+5{\cdot}9604\ 6447\ 754_{10} - 12$
$+9{\cdot}9999\ 9591\ 715_{10} - 1$	$+2{\cdot}3841\ 8579\ 102_{10} - 12$
$-9{\cdot}9999\ 9842\ 226_{10} - 1$	$+2{\cdot}3841\ 8579\ 102_{10} - 12$
	$+2{\cdot}3841\ 8579\ 102_{10} - 12$

It is evident that the residuals are as small as can be obtained using a 39-bit mantissa and the solutions themselves are best possible having regard to the condition of A. With a tolerance of 10^{-4} the rank was given as 4 and the residuals were of order 10^{-7} and 10^{-5} for the right-hand sides b_1 and b_2 respectively. This behaviour is fairly typical for an ill-conditioned A corresponding to compatible right-hand sides with solutions which are comparable in size with $||b||/||A||$.

The same problem was solved using the factorisation given by equation (8.1) which is an elimination technique. The solutions were

$$A = \begin{bmatrix} 3{\cdot}60360 & 1{\cdot}80180 & 1{\cdot}20120 & 0{\cdot}90090 & 0{\cdot}72072 & 0{\cdot}60060 \\ 1{\cdot}80180 & 1{\cdot}20120 & 0{\cdot}90090 & 0{\cdot}72072 & 0{\cdot}60060 & 0{\cdot}51480 \\ 1{\cdot}20120 & 0{\cdot}90090 & 0{\cdot}72072 & 0{\cdot}60060 & 0{\cdot}51480 & 0{\cdot}45045 \\ 0{\cdot}90090 & 0{\cdot}72072 & 0{\cdot}60060 & 0{\cdot}51480 & 0{\cdot}45045 & 0{\cdot}40040 \\ 0{\cdot}72072 & 0{\cdot}60060 & 0{\cdot}51480 & 0{\cdot}45045 & 0{\cdot}40040 & 0{\cdot}36036 \\ 0{\cdot}60060 & 0{\cdot}51480 & 0{\cdot}45045 & 0{\cdot}40040 & 0{\cdot}36036 & 0{\cdot}32760 \\ 0{\cdot}51480 & 0{\cdot}45045 & 0{\cdot}40040 & 0{\cdot}36036 & 0{\cdot}32760 & 0{\cdot}30030 \end{bmatrix}$$

Fig. 1

1st solution	*residual*
$+9{\cdot}9999\ 9995\ 087_{10} - 1$	$+4{\cdot}2915\ 3442\ 383_{10} - 11$
$+1{\cdot}0000\ 0011\ 147_{10} + 0$	$-5{\cdot}9604\ 6447\ 754_{10} - 12$
$+9{\cdot}9999\ 9351\ 852_{10} - 1$	$+1{\cdot}7881\ 3934\ 326_{10} - 11$
$+1{\cdot}0000\ 0150\ 908_{10} + 0$	$+7{\cdot}1525\ 5737\ 305_{10} - 12$
$+9{\cdot}9999\ 8470\ 583_{10} - 1$	$+3{\cdot}5762\ 7868\ 652_{10} - 12$
$+1{\cdot}0000\ 0056\ 296_{10} + 0$	$+2{\cdot}3841\ 8579\ 102_{10} - 12$
	$+8{\cdot}9406\ 9671\ 631_{10} - 12$

2nd solution	*residual*
$+1{\cdot}0000\ 0000\ 614_{10} + 0$	$+1{\cdot}0728\ 8360\ 596_{10} - 11$
$-1{\cdot}0000\ 0018\ 578_{10} + 0$	$+5{\cdot}9604\ 6447\ 754_{10} - 12$
$+1{\cdot}0000\ 0130\ 085_{10} + 0$	$+4{\cdot}7683\ 7158\ 203_{10} - 12$
$-1{\cdot}0000\ 0346\ 134_{10} + 0$	$+2{\cdot}3841\ 8579\ 102_{10} - 12$
$+1{\cdot}0000\ 0388\ 251_{10} + 0$	$-3{\cdot}5762\ 7868\ 652_{10} - 12$
$-1{\cdot}0000\ 0154\ 821_{10} + 0$	$-1{\cdot}1920\ 9289\ 551_{10} - 12$
	$+1{\cdot}0728\ 8360\ 596_{10} - 11$

The results given by the two procedures are of comparable accuracy although an elimination type method used with the normal equations gives no accuracy.

As an example of the use of the procedures with an incompatible right-hand side, b_3 was taken to be $3{\cdot}60360e_7$ where e_7 is the last column of I_7 so that the correct solution is the last column of the generalised inverse of the Hilbert segment itself. The results with the two procedures were

Orthogonalisation

solution	*residual*
$-1{\cdot}9648\ 8769\ 283_{10} + 3$	$+1{\cdot}6250\ 0000\ 000_{10} - 4$
$+5{\cdot}6763\ 0675\ 915_{10} + 4$	$-6{\cdot}5112\ 5000\ 000_{10} - 3$
$-3{\cdot}8698\ 1935\ 916_{10} + 5$	$+6{\cdot}5167\ 5000\ 000_{10} - 2$
$+1{\cdot}0119\ 4215\ 077_{10} + 6$	$-2{\cdot}6065\ 4375\ 000_{10} - 1$
$-1{\cdot}1213\ 5709\ 540_{10} + 6$	$+4{\cdot}8873\ 6562\ 500_{10} - 1$
$+4{\cdot}4317\ 9283\ 273_{10} + 5$	$-4{\cdot}3008\ 3125\ 000_{10} - 1$
	$+1{\cdot}4336\ 4375\ 000_{10} - 1$

Elimination method

solution	*residual*
$-1{\cdot}9648\ 9207\ 402_{10} + 3$	$+1{\cdot}5687\ 5000\ 000_{10} - 4$
$+5{\cdot}6763\ 1885\ 369_{10} + 4$	$-6{\cdot}5162\ 5000\ 000_{10} - 3$
$-3{\cdot}8698\ 2737\ 041_{10} + 5$	$+6{\cdot}5167\ 5000\ 000_{10} - 2$
$+1{\cdot}0119\ 4420\ 431_{10} + 6$	$-2{\cdot}6065\ 7500\ 000_{10} - 1$
$-1{\cdot}1213\ 5933\ 785_{10} + 6$	$+4{\cdot}8873\ 6250\ 000_{10} - 1$
$+4{\cdot}4318\ 0159\ 613_{10} + 5$	$-4{\cdot}3008\ 4062\ 500_{10} - 1$
	$+1{\cdot}4335\ 7812\ 500_{10} - 1$

The solutions agree to almost six figures, the maximum agreement that can be expected having regard to the condition of A. The residual should not, of course, be zero and should be in the direction orthogonal to the six columns of A. In fact it makes an angle of 10^{-6} with this direction.

Acknowledgements

The authors wish to thank Dr. D. W. Martin for reading this paper and making many valuable suggestions. The work described above has been carried out at the National Physical Laboratory.

References

*Ben-Israel, A., and Charnes, A. (1963). Contributions to the theory of generalized inverses, *J. SIAM*, Vol. 11, pp. 667–699.

*Ben-Israel, A., and Wersan, S. J. (1963). An elimination method for computing the generalized inverse of an arbitrary complex matrix, *JACM*, Vol. 10, pp. 532–537.

Givens, J. W. (1958). Computation of plane unitary rotations transforming a general matrix to triangular form, *J. SIAM*, Vol. 6, pp. 26–50.

Golub, G. H. (1965). Numerical methods for solving linear least squares problems, *Numer. Math.*, Vol. 7, pp. 206–216.

*Golub, G. H., and Kahan, W. (1964). Calculating the singular values and pseudo-inverse of a matrix, Stanford Technical Report CS8.

*Golub, G. H., and Reinsch, C. (1970). Singular value decomposition and least squares solutions, *Numer. Math.*, Vol. 14, pp. 403–420.

Householder, A. S. (1958). Unitary triangularization of a nonsymmetric matrix, *JACM*, Vol. 5, pp. 339–342.

*Noble, B. (1966). A method for computing the generalized inverse of a matrix, *SIAM J. Num. Anal.*, Vol. 3, pp. 582–584.

*Penrose, R. (1955). A generalized inverse for matrices, *Proc. Camb. Phil. Soc.*, Vol. 51, pp. 406–413.

Powell, M. J. D. (1969). A FORTRAN subroutine to invert a rectangular matrix of full rank, U.K.A.E.A. Research Group Report, AERE–R 6072.

*Rust, B., Burruss, W. R., and Schneeberger, C. (1966). A simple algorithm for computing the generalized inverse of a matrix, *CACM*, Vol. 9, pp. 381–387.

*Tewarson, R. P. (1968). A computational method for evaluating generalized inverses, *The Computer Journal*, Vol. 10, pp. 411–413.

Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem*, Oxford University Press (London).