N.T.P.

SOLUTION OF LINEAR SYSTEMS OF HIGH ORDER

This report was prepared in accordance with Contract NORD-9596 between the Bureau of Ordnance, Navy Department and the Institute for Advanced Study.

V. Bargmann
D. Montgomery
J. von Neumann

25 October 1946

SOLUTION OF LINEAR SYSTEMS OF HIGH ORDER

Chapter I - Survey of Methods

1. Introduction. In many problems in applied mathematics it is necessary to solve a system of several equations in several unknowns. Although many of these systems are non-linear, it is nevertheless true that the linear case is of great interest and importance. It is the simplest case and many problems lead directly to it. Aside from this it is important because an approximate solution to a non-linear problem can often be improved by solving a system of linear equations.

Linear equations arise in statistics, in electrical networks, in approximating the solutions of linear differential and integral equations, and in many other places. The examples which we have mentioned and others are such that n, the number of equations and unknowns is often quite large. In order to obtain a fairly accurate approximation to the solution of a partial differential equation it may be necessary to choose n=20, 50, 100, or even larger, and equally large values of n may be expected to arise in other problems.

The number of elementary steps necessary to solve a linear system for values of n of this size is so great that the solution can only be carried out conveniently by means of a high speed computing machine. This will become clear from the discussion below.

In the present report we first review some of the methods available for the solution of such problems, particularly as regards the number of elementary steps involved and make a few remarks about the accuracy of the results obtained. We shall be led to suggest a variation of a known iterative process as a very satisfactory method. Although it leads to a large

number of elementary steps it meets the requirements of stability and accuracy, and it appears practical to use this method for very large values of n with the help of a high speed machine. The error in the method can be discussed with completeness, as we shall show, and to obtain accuracy to any particular number of significant figures may take a great deal less work by this method than by what a priori seems to be a simpler method, because in this "simpler" method the control of accuracy may not be so favorable.

The number of elementary steps is by no means the only criterion for the amount of labor involved in the solution of a linear system. Where there is poor control of accuracy, as in many methods, it may be necessary to carry several times as many digits in the computation as are wanted in the final result. The number of extra digits required is greatly cut down by the iterative method which we shall discuss.

For our discussion, elementary operations consist of addition, subtraction, multiplication, and division. On all known digital computing machines the time required for a multiplication or a division is much greater than that required for an addition or subtraction. It is therefore reasonable and customary in estimating the total time required for a given computation to merely enumerate the multiplications and divisions.

In connection with the solution of linear systems, Hotelling has written a useful article which summarizes a number of methods and adds new results. The exact reference is Hotelling, "Some New Methods in

Matrix Calculation", Annals of Mathematical Statistics, Vol. 14, (1943), pp. 1-34. This paper also includes a bibliography.

2. The Elimination Method. Suppose there is given the following system of equations

$$
\begin{aligned}
&a_{11}x_1 + \cdots\cdots\cdots\cdots + a_{1n}x_n = b_1 \\
&\quad\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&a_{n1}x_1 + \cdots\cdots\cdots\cdots + a_{nn}x_n = b_n
\end{aligned}
\tag{2.1}
$$

We shall describe the method of elimination and calculate the number of multiplications and divisions involved. As a first step in the elimination method we form the following equations:

$$a_{11}x_1 = b_1 - \sum_{i=2}^{n} a_{1i}x_i \tag{2.2}$$

$$\sum_{i=2}^{n}\left(a_{ji} - \frac{a_{1i}a_{j1}}{a_{11}}\right)x_i = b_j - \frac{a_{j1}}{a_{11}}b_1 \tag{2.3}$$

Renaming the coefficients, (3) may be rewritten

$$\sum_{i=2}^{n} a'_{ji}x_i = b'_j \qquad j = 2, \ldots\ldots, n \tag{2.3'}$$

We make the following table to help us in calculating the number of steps involved in forming the system (2.3) or (2.3') from (2.1)

step involved	divisions	multiplications
$1/a_{11}$	1	0
a_{j1}/a_{11}	0	n-1
$(a_{j1}/a_{11})\, b_1$	0	n-1

step involved	divisions	multiplications
$a_{1i}(a_{j1}/a_{11})$ ·	0	$(n-1)^2$
TOTAL	1	n^2-1

We then start from the system (2.3') and carry out a similar step which we see involved 1 division and $(n-1^2-1$ multiplications. Proceeding inductively, we reach an equation

$$a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \tag{2.4}$$

and we see that to reach (4) has taken (n-1) divisions and $[(n^2-1) + \ldots + (4-1)]$ multiplications. To continue with the method of elimination we must now gradually work back and obtain a value for each x_i. As a first step we obtain

$$x_n = (1/a_{nn}^{(n-1)})\, b_n^{(n-1)}$$

which takes

1 division and 0 multiplications

The remaining steps are indicated below together with the associated number of operations.

$$x_{n-1} = 1/a_{n-1,n-1}^{(n-2)} \left(b_{n-1}^{(n-2)} - a_{n-1,n}^{(n-2)}\, x_n\right) \quad \text{2 multiplications}$$

. . . . . .

. . . . . .

$$x_1 = 1/a_{11} \left(b_1 - \sum_{j=2}^{n} a_{1i}x_i\right) \quad \text{n multiplications}$$

We see that in all we have used

n divisions

and a number of multiplications equal to

$$\frac{n(n+1)\ (2n+1)}{6} - n + \frac{n(n+1)}{2} - 1 = \frac{(n^2-1)\ (n+3)}{3}$$

If we agree that divisions take an amount of time comparable to multiplications or at any rate that the time for n divisions is negligible compared to that for $n^3/3$ multiplications we may sum up the above discussion by saying that the time required for carrying out the elementary operations of this method is about

$$n^3/3 \text{ multiplication times.}$$

What we have said above refers to the solution of the system (2.1). However, a matrix may also be inverted by the elimination method and as the number of elementary operations required is then different, we make the count for this case below. For this purpose suppose we are given the system,

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n &= y_1 \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot & \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot & \\ a_{n1}x_1 + \ldots + a_{nn}x_n &= y_n \end{aligned} \tag{2.5}$$

and that we want to solve so as to express the x's in terms of the y's. We first form the equations

$$a_{11}x_1 = y_1 - \sum_{i=2}^{n} a_{1i}x_i \tag{2.6}$$

$$\sum_{i=2}^{n}\left(a_{ji} - \frac{a_{1i}a_{j1}}{a_{11}}\right)x_i = y_j - \frac{a_{j1}}{a_{11}}y_1 \tag{2.7}$$

The differences between this case and the preceding one come about because of the fact that the y's are variable and no arithmetic operations are performed upon them. Therefore in the step which eliminates x_1 there are (n-1) fewer multiplications than before, those of line 3 of the table being ommitted. Hence in this first step the count is

1 division and n(n-1) multiplications.

After $x_1,\ldots,x_k$ have been eliminated, we obtain a system in which the left hand side contains (n-k) equations and the right contains expressions of which the one in the i^{th} line is as follows:

$$y_i - a_{il}^{(k)}y_1 - \ldots\ldots\ldots\ldots - a_{ik}^{(k)}y_k$$

In eliminating x_{k+1} we need one divisions and in anology with the first step we need (n-k)(n-k-1) multiplications. Because of the changed form of the right hand side we need an additional k(n-k-1) multiplications. Hence the totals for this step are

1 division and n(n-k-1) multiplications,

where k runs from 0 to n-2. By the last elimination we arrive at an equation of the form

$$a_{nn}^{(n-1)}x_n = y_n - \sum_{i=1}^{n-1} a_{ni}^{(n-1)}y_i$$

In order to obtain x_n as a function of the y's we must perform

1 division and (n-1) multiplcations.

By the elimination we have obtained equations of which a typical one is the following.

$$x_k = 1/a_{kk}^{(k-1)}\left(y_k - \sum_{i=1}^{k-1} a_{ki}^{(k-1)}y_i - \sum_{i=k+1}^{n} b_{ki}^{(k-1)}x_i\right)$$

where $1/a_{kk}^{(k-1)}$ as well as $a_{ki}^{(k-1)}$ and $b_{ki}^{(k-1)}$ have been computed. Hence after x_{k+1} has been obtained as a linear expression in the y's, then $x_{k+1},\ldots,x_n$ must be inserted in the preceding expression. The number of multiplications involved in obtaining x_k is

$$(k-1) + (n-k)(n+1)$$

so that altogether in the first series of operations and in the second the combined totals are n divisions and $(1+2+\ldots+n-1)(n+1+n+1) = n(n^2-1)$ multiplications

If we count a division as a multiplication this is

n^3 multiplications

and in any case this is certainly the correct order of magnitude.

As far as the number of multiplications is concerned this is a very efficient method of inverting a matrix because it requires no more multiplications than are required in forming the product of two matrices which should be considered a much simpler problem. By this method the inversion of a matrix requires only three times as many multiplications as the solution of a system of equations. In many cases it is necessary to solve a number of systems of equations having the same coefficients a_{ij} but different constants on the right hand sides. If the number of such systems is sufficiently large it would be more economical to solve them by first computing the inverse and then by operating on the vector $b_1,\ldots,b_n$ with this inverse. The process of operating on $b_1,\ldots,b_n$, with the inverse involves n^2 multiplications. If the number of systems is k, the total number of operations needed when inversion is performed first is

$$n^3 + kn^2$$

whereas the number of operations in a direct solution of each system is about

$$k(n^3/3)$$

Hence the process of inverting first has, in general, the fewer number of steps if k is at least 4.

In practice, elimination would not be carried out quite as we have described it because at any step of the elimination the coefficient in the upper left hand corner might be zero or very small. In any case it would be natural to rearrange the equations so that as large a coefficient as possible appears in the upper left hand corner. This would not affect the number of multiplications involved; however it would increase the complexity of the logical control in any machine computation. We shall later make a few comments about accuracy in the elimination method.

3. Partitioning of Matrices. We shall now describe a method of inverting a matrix based on successive partitioning into matrices of lower order and we shall compute the number of divisions and multiplications involved. This method can be considered as a generalization of the elimination method and it has been discussed by authors including Hotelling [loc. cit.]

If an n by n matrix is given we may express it in the form

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix}$$

where A, B, C,D, are of type (p,p), (p,q), (q,p) and (q,q); a matrix is said to be of type (p,q) if it has p rows and q columns. Clearly $p+q=n$.

In order to invert we look for matrices U, V, X, Y of the same types such that

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} \begin{vmatrix} U & V \\ X & Y \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

This means that the following conditions must hold.

(a) $AU + BX = 1$ (c) $AV + BY = 0$

(b) $CU + DX = 0$ (d) $CV + DY = 1$

We assume that D has an inverse. Then from (b)

(e) $X = -(D^{-1}C)U$

By (a)

(f) $(A-BD^{-1}C)U = 1$

In order to avoid the computation of A^{-1} we use (d) to obtain

(g) $Y = D^{-1} - (D^{-1}C)V$

and inserting in (c) gives

(h) $(A-BD^{-1}C)V + BD^{-1} = 0$

The things which must be computed are, then the following:

I) D^{-1}

II) $J=D^{-1}C$

III) $I=BD^{-1}$

IV) $K=A-IC=A-BJ$

V) $U=K^{-1}$

VI) $X=-JU$

VII) $V=-UI$

VIII) $Y=D^{-1}-JV=D^{-1}-XI$

Let μ_n be the number of multiplications and δ_n the number of divisions involved. Divisions can occur only in I) and V). With regard to multiplications we remark that the formation of the product of two rectangular matrices of types (p,q) and (q,r) involves pqr multiplications. If we denote by μ_p, δ_p and μ_q, δ_q the multiplications and divisions used to compute D^{-1} and K^{-1} we then obtain

$$\mu_n = \mu_p + \mu_q + 3pq^2 + 3p^2q; \qquad \delta_n = \delta_p + \delta_q$$

If D^{-1} and K^{-1} are computed by an analogous partition method, then a corresponding set of equation will hold for μ_p, δ_p and μ_q, δ_q.

For n=1, $\mu_1 = 0$ and $\delta_1 = 1$. If we make the inductive assumption that $\delta_p = p$ and $\mu_p = p^3-p$, then we can conclude for all n that, independent of the mode of partitioning we have

$$\delta_n = n \qquad , \quad \mu_n = n^3-n$$

The ordinary elimination method described in the previous section arises when p = n-1 and q=1 and when, furthermore, the successive inverses are computed by a similar type of partition.

4. The Orthogonalizing Process. Another method which can be used to invert a matrix is the Gram-Schmidt orthogonalizing process. If we denote by

$$\vec{a}_1, \ldots\ldots, \vec{a}_n$$

the rows of the matrix (a_{ij}), we first must obtain n vectors

$$\vec{w}_1,\ldots,\vec{w}_n$$

which are orthogonal to each other and are of the form

$$\begin{aligned}
\vec{w}_1 &= \vec{a}_1 \\
\vec{w}_2 &= \beta_{21}\vec{a}_1 + \vec{a}_2 \\
&\cdots \\
\vec{w}_k &= \beta_{k1}\vec{a}_1 + \ldots + \beta_{k,k-1}\vec{a}_{k-1} + \vec{a}_k \\
&\cdots
\end{aligned} \tag{4.1}$$

In order to avoid the computation of square roots these vectors have not been normalized, and the square norm of w_k $(=w_{k1}^2+\ldots+w_{kn}^2)$ is equal to a number σ_k which is computed during the process. Then if W is the matrix (w_{ij})

$$W=S.V \tag{4.2}$$

where V is orthogonal and S is the diagonal matrix with elements $\sqrt{\sigma}_1,\ldots,\sqrt{\sigma}_n$. From (4.1) it follows that

$$W=BA \tag{4.3}$$

where B is a triangular matrix with 1's on the main diagonal and elements β_{ij} below the main diagonal. From (4.3)

$$A=B^{-1}W$$

and hence

$$A^{-1} = W^{-1}B.$$

From (4.2)

$$W^{-1} = V^{-1}S^{-1} = V^*S^{-1}$$

where V^* is the transpose of V. By (4.2)

$$V^* = W^*S^{-1}.$$

Inserting this in the preceding equations we finally obtain

(4.4) $$A^{-1} = W^{*}S^{-2}B.$$

The Matrix S^{-2} is a diagonal matrix with the elements

$$1/\sigma_1, \ldots\ldots, 1/\sigma_n .$$

Thus we see that no square roots are necessary.

We proceed now to examine the process in more detail and to enumerate the multiplications and divisions which are involved. For any two vectors $\vec{X}$, $\vec{Y}$ define

$$(\vec{X}, \vec{Y}) = \sum_{i=1}^{n} X_i Y_i .$$

Then the Gram-Schmidt process is explicitly defined by the following formulas.

$$\vec{w}_1 = \vec{a}_1 \quad , \sigma_1 = (\vec{w}_1, \vec{w}_1) \quad , \quad \vec{z}_1 = (1/\sigma_1)\vec{w}_1$$

$$\vec{w}_2 = \vec{a}_2 - (\vec{a}_2, \vec{z}_1)\vec{w}_1 \quad , \sigma_2 = (\vec{w}_2, \vec{w}_2) \quad , \quad \vec{z}_2 = (1/\sigma_2)\vec{w}_2$$

(4.5) $$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\vec{w}_{k+1} = \vec{a}_{k+1} - \sum_{i=1}^{k} (\vec{a}_{k+1}, \vec{z}_i)\vec{w}_i , \sigma_{k+1} = (\vec{w}_{k+1}, \vec{w}_{k+1}) ,$$

$$\vec{z}_{k+1} = (1/\sigma_{k+1})\vec{w}_{k+1}$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

We might proceed simply by orthogonalizing the a's in the order in which they are given. In the interest of stability, and at the expense of a more elaborate logical control, it appears necessary to select that a_i for orthogonalization which leads to the largest value of σ_i so that division will be by numbers as large as possible.

From (4.5) we have

(4.6) $$\sigma_{k+1} = (\vec{w}_{k+1}, \vec{w}_{k+1}) = (\vec{a}_{k+1}, \vec{a}_{k+1}) - \sum_{i=1}^{k} \sigma_i (\vec{a}_{k+1}, \vec{z}_i)^2$$

The first step in the computation should be to find

$$\alpha_s = (\vec{a}_s, \vec{a}_s) \qquad (s = 1,2,..,n)$$

which involves

$$n^2 \text{ multiplications.}$$

After the k^{th} step compute

$$(\vec{a}_{k+s}, \vec{z}_k) = \bar{\beta}_{k+s,k} \qquad (s=1,.....,n-k)$$

which requires

$$n(n-k) \quad \text{multiplications.}$$

Next compute

$$\sigma_k \bar{\beta}^2_{k+s,k} \qquad (s=1,....,n-k)$$

which requires

$$2(n-k) \quad \text{multiplications.}$$

Then compute

$$\gamma_{k+s,k} = \alpha_{k+s} - \sum_{i=1}^{k} \sigma_i \bar{\beta}^2_{k+s,i}$$

which requires no further multiplications. Choose the largest $\gamma_{k+s,k}$, which occurs say for $s=s_o$ and form

$$\vec{w}_{k+1} = \vec{a}_{k+s_o} - \sum_{i=1}^{k} \bar{\beta}_{k+s_o,i} \vec{w}_i$$

$$\sigma_{k+1} = \gamma_{k+s_o,k}$$

$$\vec{z}_{k+1} = (1/\sigma_{k+1}) \vec{w}_{k+1}$$

which requires

$$1 \text{ division and } (k+1)n \text{ multiplications.}$$

We assume that a_{k+s_o} now becomes the vector in the position (k+1) and that the remaining ones are left in their original order.

Adding all the multiplications together we obtain

(4.7) $n^2(n+1) + n(n-1)$ multiplications

If we follow the Gram-Schmidt process without rearrangement it can be seen that the number of multiplications is

$$n^2(n+1).$$

The number of additional multiplications in the more stable procedure is n^2-n which is negligible.

If the coefficients β are computed up to the k^{th} step, then by inserting the corresponding expression (4.1) in the expression for $\vec{w}_{k+1}$ in (4.5) we obtain the coefficients $\beta_{k+1,1}, \ldots\ldots, \beta_{k+1,k}$. In computing the coefficients β as just described, let us assume that they are computed up to the k^{th} step. We then observe that to obtain the β's in the $(k+1)^{th}$ step we must have

$$\frac{k(k-1)}{2} \quad \text{multiplications}$$

that is a total of

(4.8) $$\frac{n(n-1)(n-2)}{6} \quad \text{multiplications.}$$

We have now calculated all the multiplications and divisions used in finding the matrices W , S^{-2}, and B. According to (4.4) what we must now calculate in addition is the number of multiplications necessary to form the product of these three matrices. Forming the product of S^{-2} and B requires (since the diagonal terms of B are all one)

(4.9) $$\frac{n(n-1)}{2} \quad \text{multiplications.}$$

Then forming the product of W* and $S^{-2}B$ requires

(4.10) $$\frac{n^2(n+1)}{2} \quad \text{multiplications.}$$

Adding (4.7), (4.8), (4.9), and (4.10) and neglecting quantities of lower order we see that the number of multiplications required to invert a matrix by this process is about

(4.11) $(5/3)n^3$

The number of divisions is negligible.

Although the number of multiplications is somewhat higher than in some of the other methods it is not greatly so.

5. The Method of Determinants. The solution of linear equations, or the inversion of a matrix, by the computation of determinants is not a practical method. The direct calculation of a determinant of order n from its algebraic definition requires $(n!)(n-1)$ multiplications. There exist short cut methods of calculating determinants as for example the Doolittle method which is essentially equivalent to the elimination method described above, and therefore requires about $n^3/3$ multiplications. In solving a linear system n+1 determinants must be computed, giving therefore a total of about $n^4/3$ multiplications which shows that the method is definitely inferior to the elimination method. For inverting a matrix, the method of determinants compares still more unfavorably to the elimination method.

6. Remarks on Instability. A method of computation is called stable if the rounding errors, which are inevitable in any one stage, do not tend to accumulate in a serious way. Very little is known about the stability of the methods so far described, and it appears to be difficult to obtain accurate estimates of the errors involved. What information there is tends to indicate that these methods are unstable and that rounding errors accumulate so seriously that the methods are impractical for large values of n. The number of multiplications involved is large but not excessive for an electronic machine. The real difficulty is that instability, or at least lack of information on stability, is so great, that it becomes necessary to carry large numbers of digits over

the number required in the answer. This large number of extra digits increases the amount of work to the point where it becomes prohibitive.

In considering errors it is essential to make a sharp distinction between two different types. The first type of error arises from the fact that the given data may not be exact. The error from this cause is by definition the difference between two rigorous solutions, one with the exact data, and one with inexact data. It does not depend on the method of solution, but only on the given data. It can be discussed quite accurately, and in general is far less serious than the second type of error.

The second type of error is that arising from the nature of the method of solution, being the result of an accumulation of round off errors. This type of error is very serious and must be kept under strict control in order to have any confidence at all in the final results. It depends heavily on the method used and no method can be regarded as satisfactory unless it includes a rigorous discussion of this type of error.

In the elimination method a series of n compound operations is performed each of which depends on the preceding. An error at any stage affects all succeeding results and may become greatly magnified; this explains roughly why instability should be expected. It should be noticed that at each step a division is performed by a number whose size cannot be estimated in advance and which might be so small that any error in it would be greatly magnified by division. In fact such small divisors must occur if the determinant of the matrix is small and may occur even if it is not. The divisors used are essentially principal minors of the original determinant, and even when that is safely away from zero, some of the principal minors may not be. Another reason to expect instability is

that once the variable x_n is obtained all the other variables are expressed in terms of it.

Hotelling (loc.cit.) has given a rough estimate of the error to be expected in the elimination method for a special class of matrices, namely statistical correlation matrices. For this case he has estimated that the error may be magnified by a factor whose order is 4^n. Assuming this to be correct we see that to obtain accuracy to k digits it would be necessary to use

$$n \log_{10} 4 + k \approx .6\, n + k$$

digits in the computation.

As we shall remark later it is sufficient to consider positive definite matrices (at the expense of two matrix multiplications), and it is likely that restriction to such matrices will yield more favorable estimates in certain cases. It is also likely that the method of partitioning would yield more favorable estimates. Nevertheless, it is reasonable to expect that stability under the methods so far discussed will not be as good as in the iterative method which we discuss below. Forming the determinant from the definition, aside from the fact that the amount of labor involved is prohibitive, would appear to be a quite particularly unstable process.

It is rather natural to turn to a method of successive approximations, for by its very nature such a method has the advantage that rounding errors are not very serious. This is because the rounding errors in each stage of the process are not combined with those in preceding stages. Furthermore we shall show below that it is possible to estimate accurately what the errors will be, and it will turn out that the number of extra digits which must be carried is not excessive.

Chapter II - Theoretical Determination of Maximal and Minimal Proper Values of Symmetric Positive Definite and Positive Semi-definite Matrices

7. Reduction of Inversion of Matrices to the Case of Symmetric Positive Definite Matrices. A symmetric matrix A is called positive definite if the quadratic form $(\vec{Ax},\vec{x})$ is positive unless $\vec{x}$ vanishes, and it is called positive semi-definite if $(\vec{Ax},\vec{x})$ is always non-negative. The first condition is equivalent to requiring all proper values to be positive; the second is equivalent to requiring all proper values to be non-negative. In the sequel whenever the terms definite and semi-definite occur, they will mean positive definite and positive semi-definite.

We shall show why, in the study of linear systems, it is sufficient and even desirable to consider only symmetric positive definite matrices. Suppose there is given the following arbitrary system

$$\vec{Mx} = \vec{y}$$

where M is an arbitrary non singular n by n matrix and x and y are vectors with n components. Then

$$M^*\vec{Mx} = M^*\vec{y}$$

where M^* is the transpose of M, and we know that M^*M is symmetric and positive definite. The second set of equations is equivalent to the first and hence a knowledge of how to treat the second case will enable us to treat the non-symmetric case. The same is true for the problem of inverting a matrix as we see from the following equations:

$$(M^*M)^{-1} = M^{-1}M^{*-1}$$

and

$$M^{-1} = (M^*M)^{-1}M^*$$

Thus once $(M^*M)^{-1}$ is obtained, one additional matrix multiplication yields M^{-1}.

One advantage of using symmetric matrices is that in squaring such matrices it is only necessary to compute the elements in the main diagonal and above. More generally, the same is true in multiplying two commutative symmetric matrices, and only such matrix multiplications occur in the process to be proposed, since the result is sure to be symmetric. Hence we need make only $\frac{n^2(n+1)}{2}$ multiplications instead of the n^3 which are required in general. In addition, positive definite symmetric matrices have other properties which are very convenient. For example the proper values are positive real numbers and the largest one of these is the bound of the matrix.

A matrix A can be inverted only if it is non-singular, that is if none of its proper values is equal to zero. In computation, numbers become distorted by round off errors and hence in the case of practical computation the question is not whether some proper value is zero but rather whether some proper value is smaller than some given small quantity ε. In the case of a symmetric matrix this is equivalent to the question of whether or not the bound of the inverse is at most $1/\varepsilon$.

In finding the inverse of a matrix A by iterative procedures there are two general methods either of which might be followed. First a process might be devised which would converge to the inverse when its bound is less than $1/\varepsilon$, and otherwise would yield a sequence of matrices B_k whose bounds increase beyond $1/\varepsilon$ and such that B_kA does not come near 1. This method would then either yield the inverse or the information that the bound of the inverse is greater than $1/\varepsilon$, and that consequently the inverse can not be obtained with the number of digits being used. A second method would be to estimate in advance the value of the minimum proper value, and then to use this information to decide whether the inverse can be computed, and also to speed the computation. It is this seoond method which will mainly

concern us, and we shall discuss practical methods of obtaining maximal and minimal proper values. These methods are also of interest for their own sake.

8. Summary of Facts about Matrices. For any vector $\vec{x}$ we define

$$|\vec{x}| = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}$$

Clearly

$$(\vec{x},\vec{x}) = |\vec{x}|^2$$

and by Schwarz's well known inequality $|(\vec{x},\vec{y})| \leq |\vec{x}|\,|\vec{y}|$.

We define next for any matrix A the trace as follows

$$t(A) = \sum_{i=1}^{n} a_{ii}$$

and norm as

$$NA = [t(A^*A)]^{\frac{1}{2}} = [t(AA^*)]^{\frac{1}{2}} = [\sum_{ij} a_{ij}^2]^{\frac{1}{2}} .$$

The bound of A is given by

$$|A| = \operatorname*{Max}_{|\vec{x}|=1} \frac{|A\vec{x}|}{|\vec{x}|}$$

so that, clearly,

$$|A| = \operatorname*{Max}_{\vec{x}\neq 0} \frac{|A\vec{x}|}{|\vec{x}|} .$$

Furthermore for $\vec{x}\neq 0$, $\vec{y}\neq 0$

$$\frac{|(A\vec{x},\vec{y})|}{|\vec{x}|\,|\vec{y}|} \leq \frac{|A\vec{x}|\,|\vec{y}|}{|\vec{x}|\,|\vec{y}|} = \frac{|A\vec{x}|}{|\vec{x}|}$$

and when $\vec{y} = A\vec{x}$

$$\frac{(A\vec{x},\vec{y})}{|\vec{x}|\,|\vec{y}|} = \frac{|A\vec{x}|^2}{|\vec{x}|\,|A\vec{x}|} = \frac{|A\vec{x}|}{|\vec{x}|} .$$

Hence

$$|A| = \operatorname*{Max}_{\vec{x}\neq 0} \frac{|A\vec{x}|}{\vec{x}} = \operatorname*{Max}_{\vec{x},\vec{y}\neq 0} \frac{|(A\vec{x},\vec{y})|}{\vec{x}\ \vec{y}} .$$

The triangular inequality (in n dimensions)

$$|\vec{x} + \vec{y}| \leq |\vec{x}| + |\vec{y}|$$

is well known. In n^2 dimensions it gives

$$N(A+B) \leq NA + NB$$

The vectorial triangular inequality gives immediately

$$\|A + B\| \leq \|A\| + \|B\| .$$

From the definitions it follows that

$$NA^* = NA .$$

We also have

$$(A^*\vec{x},\vec{y}) = (\vec{x},A\vec{y}) = (A\vec{y},\vec{x})$$

and hence the second expression for $|A|$ gives

$$|A^*| = |A| .$$

The estimate

$$|AB| \leq |A|\,|B|$$

is immediate, and iterating it gives

$$|A^s| \leq |A|^s$$

If the j^{th} column of B is denoted by u_j then the j^{th} column of AB is $v_j = Au_j$. We see that

$$NB = [\sum_j u_j{}^2]^{\frac{1}{2}} , \quad N(AB) = [\sum_j v_j{}^2]^{\frac{1}{2}}$$

and

$$|v_j| \leq |A|\,|u_j| .$$

Hence

$$N(AB) \leq |A|\,NB .$$

Replacing AB, A, B by $(AB)^* = B^*A^*$, B^*, A^* and using our previous results, transforms this into

$$N(AB) \leq |B|\,NA .$$

For any matrix A,

$$NA = N(1\cdot A) \leq N1\cdot A = n^{\frac{1}{2}} A ,$$

where 1 is the unit matrix. If $\vec{w}_j$ is the j^{th} row of A, then the vector Ax has the components $(\vec{w}_j,\vec{x})$ and

$$NA = [\sum_j |\vec{w}_j|^2]^{\frac{1}{2}} .$$

Hence

$$|A\vec{x}| = [\sum_j (\vec{w}_j, \vec{x})^2]^{\frac{1}{2}} \leq [\sum_j |\vec{w}_j|^2 |\vec{x}|^2]^{\frac{1}{2}} = NA\ |x|$$

so that

$$|A| \leq NA .$$

Summing up, we have

$$|A| \leq NA \leq n^{\frac{1}{2}} |A|.$$

We see that

$$\frac{(A^*A\vec{x}, \vec{x})}{|\vec{x}|^2} = \frac{(A\vec{x}, A\vec{x})}{|\vec{x}|^2} = \frac{|A\vec{x}|^2}{|\vec{x}|^2} .$$

Comparing the second definition of bound as applied to A*A with the original one for A gives

$$|A^*A| \geq |A|^2 .$$

However

$$|A^*A| \leq |A^*|\,|A| = |A|^2$$

and therefore

$$|A^*A| = |A|^2 .$$

If A is such that

$$a_{ij} \leq c$$

then $|t\ (A)| \leq nc$

$$NA \leq nc$$

$$|A| \leq nc$$

The equality signs in the above may hold as they do in the case where each a_{ij} is equal to c. In the case of $|A|$ we see this by applying A to the vector whose components are all unity.

In any mechanical computation it is wise, so far as possible, not to deal with numbers of different orders of magnitude. Hence in carrying out the computation of the maximal and minimal proper values it is desirable

to make sure that the matrix elements occuring are less than one in absolute value, and on the other hand do not get exceedingly small. In order to insure the first it is sufficient to keep the bounds of the matrices occuring at most one. For suppose that A is a matrix. Let $\vec{\varphi}_1, \cdots\cdots, \vec{\varphi}_n$ be the unit coordinate vectors. Then the matrix element a_{ij} is given as follows:

$$a_{ij} = (\vec{\varphi}_i, A\vec{\varphi}_j)$$

Hence

$$|a_{ij}| \leq |\vec{\varphi}_i| \cdot |A\vec{\varphi}_j| \leq |\vec{\varphi}_i| \cdot |A| \cdot |\vec{\varphi}_j| = |A|.$$

Therefore if A has bound at most one all elements occuring have absolute value at most one. From the preceding, if $|A| \leq 1$ and $|B| \leq 1$, then $|AB| \leq 1$. Thus $|A| \leq 1$ is a property hereditary under multiplication whereas the property of having elements with absolute values at most one is not hereditary, as we shall see later.

If $|A| < 1$, then 1-A has an inverse. It is sufficient to show that $(1-A)\vec{x} \neq 0$ if $\vec{x} \neq 0$. This follows from the inequality

$$|(1-A)\vec{x}| = |\vec{x}-A\vec{x}| \geq |\vec{x}| - |A\vec{x}| \geq |\vec{x}| - |A||\vec{x}| = (1-|A|)\,|\vec{x}| > 0$$

which holds for $\vec{x} \neq 0$. Furthermore in this case

$$|(1-A)^{-1}| \leq \frac{1}{1-|A|}$$

In fact we have

$$1 = (1-A)\ (1-A)^{-1} = (1-A)^{-1} - A(1-A)^{-1}$$

and hence

$$|A|\,|(1-A)^{-1}| \geq |A(1-A)^{-1}| = |(1-A)^{-1} - 1| \geq |(1-A)^{-1}| - 1$$

$$(1-|A|)\ |(1-A)^{-1}| \leq 1.$$

If A is symmetric then the relation $|A*A| = |A|^2$

becomes

$$|A^2| = |A|^2$$

Iterating this gives

$$|A^{2^t}| = |A|^{2^t}$$

Given any s, choose t with 2^t greater than s. Then

$$|A^s| \leq |A|^s$$

$$|A^{2^t-s}| \leq |A|^{2^t-s}$$

$$|A^{2^t}| \leq |A^{2^t-s}|\,|A^s| \leq |A|^{2^t-s}\,|A|^s = |A|^{2^t}.$$

The fact that the first and last terms above are equal, as has been seen, implies that all the inequality signs above are equality signs and that

$$|A^s| = |A|^s$$

for any symmetric matrix.

The definitions of bound, norm, and trace are invariant with respect to orthogonal coordinate transformations. It may therefore be assumed that a symmetric matrix is in diagonal form with proper values $\lambda_1, \ldots, \lambda_n$. It is evident that

$$t(A) = \sum_{i=1}^{n} \lambda_i$$

$$NA = \left(\sum_{i=1}^{n} \lambda_i^2\right)^{\frac{1}{2}}$$

$$|A| = \operatorname{Max}_i |\lambda_i|$$

From the last equation we again obtain a proof that $|A^s| = |A|^s$ for symmetric matrices. For a semi-definite symmetric matrix A all proper values are non-negative and hence

$$|A| \leq t(A)$$

It will now be seen that the property of having elements of absolute value at most one is not hereditary under multiplication. Let A be symmetric positive definite matrix with $|A| > 1$. Then

$$|A^s| = |A|^s$$

and $|A|^s$ tends to infinity with s. Since

$$|A^s| \leq t(A^s)$$

$t(A^s)$ also tends to infinity with s and some diagonal element must be unbounded. As we have seen, if a matrix is such that

$$|a_{ij}| \leq 1,$$

$|A|$ may be as large as n.

9. Determination of Maximum Proper Value. We now describe the process to be used in estimating the largest proper value of a semi-definite matrix A (the smallest is obtained by a very similar method as we shall see). We first neglect round off errors entirely and then later return to consider how these may be controlled.

We have already noticed that

$$|A^s| = |A|^s \tag{9.1}$$

and

$$\frac{1}{n} \leq \frac{|A|}{t(A)} \leq 1 \tag{9.2}$$

Applying this inequality to A^s we have

$$\frac{1}{n} \leq \frac{|A|^s}{t(A^s)} \leq 1$$

or taking roots

$$\frac{1}{n^{1/s}} \leq \frac{|A|}{[t(A^s)]^{1/s}} \leq 1$$

Since $n^{1/s}$ approaches 1 we see that

$$[t(A^s)]^{1/s} \longrightarrow |A|$$

The convergence can be speeded by successive squaring, that is by taking $s = 2^k$. Hence

$$n^{-2^{-k}} \leq \frac{|A|}{[t(A^{2^k})]^{2^{-k}}} \leq 1 \tag{9.3}$$

Note that

$$n^{-2^{-k}} = e^{-(\log n)2^{-k}} \geq 1 - (2^{-k}) \log n$$

so that

$$1 - 2^{-k} \log n \leq \frac{|A|}{[t(A^{2^k})]^{2^{-k}}} \leq 1$$

This formula shows that the approximation will be good for moderate values of k, because the sizes of n under consideration cause no serious difficulty. For example for $n = 100$, $\log n = 4.6$.

The above formula can be expressed by an iterative scheme:

$$A_0 = A \ , \qquad A_k = (A_{k-1})^2$$

$$m_k = [t(A_k)]^{2^{-k}}$$

and then

$$1 - 2^{-k} \log n \leq \frac{|A|}{m_k} \leq 1$$

Unless $|A| = 1$, A_k will tend either to zero or infinity and in either case the requirements we laid down for the matrix elements are violated. In order to circumvent this difficulty we could make use of the device of normalizing all matrices which occur so that their traces are 1. This would guarantee a bound between $1/n$ and 1.

Actually in order to control the computation we modify the scheme still more as we now indicate. We will choose a number r slightly less than one in a menner which will later be described exactly. This number will be so chosen that even with round off errors, bounds are < 1.

We assume that A has trace r and if not we divide by the trace and multiply by r. Then

(9.4)
$$B_0 = A \quad , \quad B_k = \frac{r}{n_k} (B_{k-1})^2$$

where

(9.5)
$$n_k = t(B_{k-1}^2)$$

Hence for all k
$$t(B_k) = r.$$

Then B_k is proportional to A_k, that is

$$A_k = c_k B_k$$

and the recursive definition gives

$$c_0 = 1 \quad , \quad c_k = \frac{n_k}{r} c_{k-1}^2$$

so that
$$c_k = \left(\frac{n_1}{r}\right)^{2^{k-1}} \left(\frac{n_2}{r}\right)^{2^{k-2}} \cdots\cdots\cdots\cdots \frac{n_k}{r}$$

Next
$$t(A_k) = rc_k$$

and hence
$$m_k = [t(A_k)]^{2^{-k}} = \left(\frac{n_1}{r}\right)^{\frac{1}{2}} \left(\frac{n_2}{r}\right)^{1/4} \cdots \left(\frac{n_k}{r}\right)^{2^{-k}} r^{2^{-k}}$$

$$= \frac{n_1^{\frac{1}{2}} \; n_2^{1/4} \cdots\cdots\cdots n_k^{2^{-k}}}{r^{1-2^{-(k-1)}}}$$

As we have seen before
$$n^{-2^{-k}} \leq \frac{|A|}{m_k} \leq 1$$

Substitution gives
$$\left(\frac{n}{r^2}\right)^{2^{-k}} \leq \frac{|A|}{1/r \; (n_1^{\frac{1}{2}} \; n_2^{1/4} \cdots n_k^{2^{-k}})} \leq r^{2^{-(k-1)}} \leq 1$$

Thus $(1/r)(n_1^{\frac{1}{2}}\, n_2^{1/4} \ldots\ldots\, n_k^{2^{-k}})$ is a very rapidly converging product development for $|A|$.

It should also be noted that $t(B_{k-1}) = r$ implies

$$r^2/n \leq t(B_{k-1}^2) \leq r$$

that is

$$r^2/n \leq n_k \leq r$$

Also $t(B_k) = r$ implies

$$r/n \leq |B_k| \leq r$$

Hence the algorithm just stated produces only quantities in the desired range, that is matrices with bound less than one and numbers of absolute value less than one.

We now add some remarks which will be useful in what follows. The procedure discussed for obtaining the bound of a matrix holds for indefinite symmetric matrices. It is true that (9.2) holds only for semi-definite matrices. However since (9.1) holds for every symmetric matrix and since in (9.3) only even powers occur we see that (9.3) is always true. The same remarks apply to the procedure defined by (9.4) and (9.5) so that (9.6) always it true. Thus if it is known for a symmetric matrix that its largest positive proper value is greater than the absolute of any negative proper value, then these procedures determine this largest proper value.

Once the maximum proper value is obtained we may proceed in the following way to obtain the minimum proper value. Let ρ be any number between the maximum proper value and 1. Then the matrix

$$\rho I - A$$

is symmetric and positive definite and we may use the previously discussed method to obtain its maximum proper value. If we subtract this from ρ we obtain the minimum proper value of A.

Chapter III - Numerical Computation of Maximum and Minimum Proper Values of A

10a. Calculation of A from M. When a matrix M is given, it is necessary to prepare it for the methods used below. In this section we discuss this preparation and the errors involved. Numbers are assumed to be given in terms of a base b which can be any integer greater than one, but which for all practical purposes will be either 2 or 10.

As we have already remarked the error in the final result of numerical work arises from two distinct causes 1) errors in the given data and 2) errors arising from the particular method used in the computation. In the succeeding sections we discuss errors of the second kind, that is we discuss the errors involved in finding the proper values or the inverse of M as though M were rigorously given; later some remarks are made about the first type of error.

The elements of M are assumed given to a certain number of significant figures. As will be shown in detail it is usually necessary to carry more figures than these in the computation so that zeros will have to be added to obtain the number of figures required in the computation.

Let p be the number of digits to be carried in the computation. Our first step is to move the decimal (or binary) point in a way we shall now indicate. Since it is desirable that all numbers occuring have absolute value less than one, we shall multiply M by b^h where h will be chosen as follows. Let β be the maximum of the elements m_{ij}. Then h is chosen such that

$$b^h \beta < (1/n)\ (1-n^2(n+1)\delta_\bullet)^{\frac{1}{2}},$$
$$b^{h+1} \beta \geq (1/n)\ (1-n^2(n+1)\delta_\bullet)^{\frac{1}{2}},$$

where $\delta_0 = (1/2)\, b^{-p}$ is the maximum round off error in a multiplication. Then the matrix

$$M_1 = b^h M \tag{10.1}$$

is formed with no errors at all simply by shifting the decimal (binary) point. All the elements of M_1 have an absolute value less than

$$\alpha = (1/n)\ (1 - n^2\,(n+1)\,\delta_0)^{\frac{1}{2}} \tag{10.2}$$

This choice of α will be justified a little later.

We wish to calculate a matrix whose mathematical definition is

$$r\,\frac{M_1^* M_1}{t(M_1^* M_1)} \tag{10.3}$$

where the following conditions are satisfied

$$\begin{aligned} &.9 \leq r \leq 1-4n^3\delta \\ &n^3\delta \leq 1/50 \\ &n \geq 3 \end{aligned} \tag{10.4}$$

As will be seen later it will be desirable to use a greater accuracy in computing $M_1^* M_1$ than in computing the largest proper value. We therefore distinguish δ_0 which is used in the first procedure from δ which is used in the second and we assume

$$\delta_0 \leq \delta$$

The number r will be kept fixed in the following discussion.

In forming $M_1^* M_1$ an error matrix X is introduced so that what is really obtained is

$$M_1^* M_1 + X \tag{10.5}$$

The numbers involved are less than one so that the multiplication of two numbers introduces a round off error of at most δ_0, and therefore each

element x_{ij} of X is such that

$$|x_{ij}| \leq n\delta_0 .$$

This shows that

$$(10.6) \qquad |X| \leq n^2\delta_0 , \quad NX \leq n^2\delta_0 , \quad t(X) \leq n^2\delta_0 .$$

The matrix

$$(10.7) \qquad M_1^* M_1 + X + n^2\delta_0 \cdot 1$$

will be definite and for convenience we work with this matrix. The addition of $n^2\delta_0 \cdot 1$ is, in any case, a small error and in finding proper values it creates no error at all since its effect is known and is to add $n^2\delta_0$ to each proper value. Possibly it is worth remarking that symmetry in the product matrix is assured by computing the elements on and above the main diagonal and defining the remaining elements accordingly.

Since $M_1^* M_1 + X + n^2\delta_0 \cdot 1$ is semi-definite, the absolute value of each element is less than the trace, and for this reason the numerical computation of

$$\frac{M_1^* M_1 + X + n^2\delta_0 \cdot 1}{t(M_1^* M_1 + X + n^2\delta_0 \cdot 1)}$$

gives a matrix

$$(10.8) \qquad \frac{M_1^* M_1 + X + n^2\delta_0 \cdot 1}{t(M_1^* M_1 + X + n^2\delta_0 \cdot 1)} + Y$$

where each element of Y has absolute value at most δ_0 so that

$$(10.9) \qquad |Y| \leq n\delta_0 , \quad NY \leq n\delta_0 , \quad t(Y) \leq n\delta_0 .$$

The last step is to multiply the matrix (10.8) by r which gives a new error matrix Z each of whose elements is at most δ_0 in absolute value, and hence

$$|Z| \leq n\delta_0 , \qquad NZ \leq n\delta_0 , \qquad t(Z) \leq n\delta_0$$

This operation, then, leads to

$$A = r\left[\frac{M_1^* M_1 + X + n^2\delta_0}{t(M_1^* M_1 + X + n^2\delta_0)} + Y\right] + Z \tag{10.10}$$

We shall first see that all the numbers occuring in the calculation of A are of absolute value smaller than one. The elements in (10.5) and (10.7) satisfy this requirement because an element of $M_1^* M_1$ is smaller than the trace in absolute value and

$$t(M_1^* M_1) < n^2\alpha^2$$

Furthermore the absolute value of every element in (10.7) is majorized by

$$t(M_1^* M_1) + NX + n^2\delta_0 < n^2\alpha^2 + 2n^2\delta_0$$

Also

$$t(M_1^* M_1 + X + n^2\delta_0 \cdot 1) < n^2\alpha^2 + n^2\delta_0 + n^3\delta_0 = 1$$

The number α was so chosen as to insure the last inequality.

We shall next consider the norm of $B_0 = A$, and we first recall that if a semi-definite matrix is divided by its trace the norm of the resulting matrix is at most one. Hence

$$NA \leq r(1 + NY) + NZ \leq r(1 + n^2\delta_0) + n^2\delta_0$$

It is easily seen from (10.4) and the above that

$$NA < \sqrt{1 - n^2\delta} \tag{10.11}$$

In particular it follows that all elements of A have absolute value smaller than 1 and this completes the proof that this is the case for all elements occuring.

By (10.10) we see that t (A) is given by

$$t(A) = r + rt(Y) + t(Z)$$

and hence

$$|t(A) - r| < 2n\delta. \tag{10.11}$$

By (10.10) A is expressed as the sum of a semi-definite matrix and the matrices rY and Z. Hence by Courant's theorem stated in section 11, all the proper values of A are larger than

$$-(r|Y| + |Z|) \geq -2n\delta. \tag{10.11b}$$

On the other hand

$$t(A) \geq r - 2n\delta.,$$

so that the largest proper value of A is at least

$$\frac{r - 2n\delta.}{n}$$

From the conditions on r we see therefore that this largest proper value is positive and greater in absolute value than any negative proper value.

10b. Control of Numbers Occuring in Computation of Largest Proper Value. In the computation of the largest proper value we shall begin with the matrix A. We wish to make the computations which are rigorously defined by the following inductive procedure

$$B_0 = A$$

$$B_{k+1} = r\frac{B_k^2}{t(B_k^2)}$$

In carrying out this procedure round off errors occur so that the actual computation will proceed as in the four steps listed below. Instead of the theoretically defined sequence B_k, we will obtain by computation a certain sequence B_k' and we now describe in detail the computation of B_{k+1}', $k = 0,1,2,\ldots$. Let $B_0' = A$.

Squaring B_k' we obtain

I. $U_{k+1} = B_k'^2 + X_{k+1}$

where X_{k+1} is the matrix of round off errors.

Then the trace is computed and is

II. $t(U_{k+1}) = t(B_k'^2) + t(X_{k+1})$

Multiplying I by r gives

III. $V_{k+1} = rU_{k+1} + Y_{k+1}$

where Y_{k+1} is the error matrix. As a final step we have

IV. $$B'_{k+1} = \frac{V_{k+1}}{t(U_{k+1})} + Z_{k+1}$$

This last step can be written in two parts

IVa. $$B'_{k+1} = r\,\frac{U_{k+1}}{t(U_{k+1})} + D_{k+1}$$

IVb. $$D_{k+1} = \frac{Y_{k+1}}{t(U_{k+1})} + Z_{k+1}$$

We know that the following two inequalities hold when k = 0.

(10.12) $$NB_k' < \sqrt{1-n\,\delta}$$

(10.13) $$r - \theta \leq t(B_k') \leq r + \theta \quad , \quad \theta = 2n^2\delta$$

We wish next to prove that if (10.12) and (10.13) hold for k, then (10.14) hold for k+1. Then (10.12) and (10.13) (i.e. (e) and (f) in (10.14)) hold for k+1 too. Hence (10.12) and (10.13) hold for all k=0,1,2,..., and so (10.14) holds for all k=1,2,....

(10.14)

(a) $NU_{k+1} < 1$, (b) $|t(U_{k+1})| < 1$,

(c) $NV_{k+1} < 1$, (d) $NV_{k+1} < t(U_{k+1})$,

(e) $NB'_{k+1} < \sqrt{1-n^2\delta}$ (f) $r - \theta \leq t(B'_{k+1}) \leq r + \theta$

Conditions (a), (b), (c), and (e) insure that all numbers occuring have absolute value less than one. Condition (d) insures that in the division,

the divisor is greater than the dividend.

By (I.) elements of X_{k+1} are, in absolute value, at most n and hence

$$|t(X_{k+1})| \leq n^2\delta\ , \quad |X_{k+1}| \leq NX_{k+1} \leq n^2\delta \tag{10.15}$$

But

$$NU_{k+1} \leq N(B_k'^2) + NX_{k+1} \leq (NB_k')^2 + NX_{k+1} < 1-n^2\delta + n^2\delta = 1$$

which proves (a)

Also

$$|t(U_{k+1})| \leq |t(B_k'^2)| + |t(X_{k+1})| = (NB_k')^2 + |t(X_{k+1})| \leq 1-n^2\delta + n^2\delta = 1$$

which proves (b).

The elements of Y_{k+1} are at most δ in absolute value, so that

$$|t(Y_{k+1})| \leq n\delta\ , \quad |Y_{k+1}| \leq NY_{k+1} \leq n\delta\ . \tag{10.16}$$

We see that condition (c) follows from (b) and (d).

Now

$$t(U_{k+1}) = t(B_k'^2) + t(X_{k+1}) \geq t(B_k'^2) - n^2\delta$$

and hence

$$\frac{NV_{k+1}}{t(U_{k+1})} \leq \frac{rNU_{k+1}}{t(U_{k+1})} + \frac{NY_{k+1}}{t(U_{k+1})}$$

$$\leq \frac{N(B_k'^2) + NX_{k+1}}{t(B_k'^2) - n^2\delta} + \frac{n\delta}{t(B_k'^2) - n^2\delta}$$

$$\leq r\ \frac{t(B_k'^2) + n^2\delta}{t(B_k'^2) - n^2\delta} + \frac{n\delta}{t(B_k'^2) - n^2\delta}$$

Let λ_i be the proper values of B_k' . Then

$$t(B_k') = \sum_{i=1}^{n} \lambda_i$$

and by (10.15)

$$t(B_k'^2) = \sum_{i=1}^{n} \lambda_i^2 \geq \frac{1}{n}\left(\sum_{i=1}^{n} \lambda_i\right)^2 = \frac{1}{n}\left(t(B_k')\right)^2 \geq \frac{(r-\theta)^2}{n}$$

Therefore

$$(10.18) \qquad p = r\,\frac{t(B_k'^2) + n^2\delta}{t(B_k'^2) - n^2\delta} \leq r\,\frac{(r-\theta)^2 + n^3\delta}{(r-\theta)^2 - n^3\delta}$$

$$(10.19) \qquad q = \frac{n\delta}{t(B_k'^2) - n^2\delta} \leq \frac{n^2\delta}{(r-\theta)^2 - n^3\delta}$$

It may be shown from our assumptions, that

$$(10.20) \qquad (r-\theta)^2 - n^3\delta > \frac{n}{2n-1} > .5$$

Hence

$$(10.21) \qquad q < 2n^2\delta$$

We shall now prove that

$$(10.22) \qquad r\,\frac{(r-\theta)^2 + n^3\delta}{(r-\theta)^2 - n^3\delta} + 2n^2\delta < 1-n^2\delta$$

This is equivalent to

$$r < (1-3n^2\delta)\,\frac{1 - \dfrac{n^3\delta}{(r-\theta)^2}}{1 + \dfrac{n^3\delta}{(r-\theta)^2}}$$

It is sufficient to prove that

$$r < (1-3n^2\delta) \cdot \left(1-2\,\frac{n^3\delta}{(r-\theta)^2}\right)$$

and since $r < 1-4n^3\delta$ it is also sufficient to prove

$$1-4n^3\delta < (1-3n^2\delta)\left(1-2\,\frac{n^3\delta}{(r-\theta)^2}\right)$$

This is equivalent to requiring

$$4n^3\delta > 2\, n^3\delta \left(\frac{1-3n^2\delta}{(r-\theta)^2} + \frac{3}{2n} \right)$$

The number in parentheses is smaller than

$$\frac{1}{2} + \frac{1}{(.88)^2} < 2$$

and this proves the inequality. From the inequality (10.22) and previous results

$$p + q < 1-n^2\delta$$

which proves

$$\frac{NV_{k+1}}{t(U_{k+1})} < 1 - n^2\delta$$

so that (d) is true.

Now that (d) has been proved it follows that every element Z_{k+1} is smaller than δ in absolute value and we have

(10.23) $$t(Z_{k+1}) \leq n\delta \quad , \quad |Z_{k+1}| \leq NZ_{k+1} \leq n\delta$$

From (10.23)

$$NB'_{k+1} \leq \frac{NV_{k+1}}{t(U_{k+1})} + NZ_{k+1} \leq 1 - n^2\delta + n\delta < \sqrt{1-n^2\delta}$$

which proves (e).

We see that

$$t\,(B'_{k+1}) = r + t\,(D_{k+1})$$

and

$$|t\,(D_{k+1})| \leq \frac{|t\,(Y_{k+1})|}{t\,(B_k'^2 + X_{k+1})} + |t\,(Z_{k+1})|$$

$$\leq \frac{n^2\delta}{(r-\theta)^2 - n^3\delta} + n\delta \leq \frac{2n-1}{n}\, n^2\delta + n\delta = 2n^2\delta .$$

Therefore $$r - \theta \leq t(B'_{k+1}) \leq r + \theta$$

which proves (f). Thus, as pointed out before (10.14), this hold for all k=1,2,....

This concludes the proof that, under the given assumptions on r, all numbers occuring are less than one in absolute value.

11. Formulation of the Method. The theoretical method of obtaining the maximum proper value has already been given, and we have also shown how the size of the numbers involved may be controlled. We turn to a discussion of the errors involved in carrying out the process numerically. The matrix A and the numbers r, n, and $n^3 \delta$ are taken subject to the same restrictions as in the preceding section.

The theoretical procedure is the following

$$B_0 = A$$

$$B_k = r \frac{B_{k-1}^2}{t(B_{k-1}^2)} \tag{11.1}$$

and we use the notation

$$n_k = t\,(B_{k+1}^2)$$

$$s_k = n_k\, n_{k-1}^2 \cdots n_1^{2^{k-1}} \tag{11.2}$$

$$m_k = s_k^{2^{-k}} = n_1^{1/2}\ \ n_2^{1/4} \cdots n_k^{2^{-k}}$$

We see that

$$s_k = n_k s_{k-1}^2 \quad , \quad n_k = \frac{s_k}{s_{k-1}^2} \tag{11.3}$$

By the numerical procedure we obtain the following

$$B_1' = B_0 = A \tag{11.4}$$

$$B'_k = r\,\frac{B'^2_{k-1} + X_k}{t(B'^2_{k-1} + X_k)} + D_k$$

and here we use the notation

$$n'_k = t\,(B'^2_{k-1} + X_k)$$

(11.5)

$$s'_k = n'_k\, n'^2_{k-1} \cdots\cdots\cdots n'^{2^{k-1}}_1$$

From this

(11.6) $$s'_k = n'_k\, s'^2_{k-1} \qquad n'_k = \frac{s'_k}{s'^2_{k-1}}$$

In what follows we shall have occasion to use the following theorem (Courant-Hilbert, Methoden der Math. Phys., Vol. I. (1931) p. 27, implies this theorem)

Theorem. Let R, S, and T be three symmetric matrices with

$$R = S + T$$

Then the proper values $\rho_1, \cdots\cdots \rho_n$ of R and $\sigma_1, \cdots\cdots \sigma_n$ of S may be arranged in such an order that

$$|\rho_i - \sigma_i| \leq |T|$$

Let $\mu_{k,1}, \cdots\cdots, \mu_{k,n}$ denote the proper values of B'_k arranged in a suitable order. Since the proper values of B'^2_{k-1} are $\mu^2_{k-1,1}, \cdots\cdots, \mu^2_{k-1,n}$ it follows from the theorem that the proper values of $B'^2_{k-1} + X_k$ are

(11.7) $$\mu^2_{k-1,i} + \varepsilon_{k,i}$$

where

$$|\varepsilon_{k,i}| \leq |X_k|$$

We have seen that every element of X_k has absolute value at most $n\delta$.

If we wish to take into account the statistical distribution of errors then for large values of n the absolute value of each element will be at most a number whose order is $\sqrt{n}\,\delta$. To avoid committing ourselves at present we assume that each element of X_k is bounded by $z\delta$, where $z \leq n$ and consequently that

$$|X_k| \leq nz\delta$$

and

$$|t(X_k)| \leq nz\delta$$

Therefore

(11.8) $$|\varepsilon_{k,i}| \leq nz\delta$$

Then by definition

(11.9) $$n'_k = \sum_{j=1}^{n} (\mu^2_{k+1,j} + \varepsilon_{k,j}).$$

From (11.7)

(11.10) $$\mu_{k,i} = r\,\frac{\mu^2_{k-1,i} + \varepsilon_{k,i}}{\sum_{j=1}^{n} (\mu^2_{k-1,j} + \varepsilon_{k,j})} + \eta_{k,i}$$

where

(11.11) $$\eta_{k,i} \leq \frac{|D_k|}{r} \leq \frac{2n^2\delta}{r}$$

For convenience we let

$$\eta_k = \sum_{i=1}^{n} \eta_{k,i} \quad , \quad \varepsilon_k = \sum_{i=1}^{n} \varepsilon_{k,i}$$

and

$$\lambda_i = \mu_{0,i} \, .$$

We take α and β such that

(11.12) $$|\eta_{k-1}| \leq \beta, \; |\eta_{k-1,i}| \leq \beta$$
$$|\varepsilon_{k-1}| \leq \alpha, \; |\varepsilon_{k-1,i}| \leq \alpha$$

These inequalities will be satisfied if $\alpha = nz\delta$ and $\beta = \frac{2n^2\delta}{r}$, because $\varepsilon_k = t(X_k)$, and $\eta_k = t\left(\frac{D_k}{r}\right)$. From the definitions,

$$t(A_k) = t(A^{2^k}) = \left(\frac{1}{r}\right)^{2^k-2} \cdot n_k n_{k-1}^2 \cdots\cdots n_1^{2^{k-1}} \tag{11.13}$$

that is

$$\sum_{i=1}^{n} \lambda_i^{2^k} r^{2^k-2} = s_k \tag{11.14}$$

What we wish to estimate

$$\frac{n'_k}{n_k} = \frac{s'_k}{s_k} \left(\frac{s'_{k-1}}{s_{k-1}}\right)^{-2} \tag{11.15}$$

We define

$$\nu_{k,i} = \mu_{k,i} \; s'_k \tag{11.16}$$

Then

$$\sum_{i=1}^{n} \nu_{k,i} = \sum_{i=1}^{n} \mu_{k,i} \; s'_k = r(1+\eta_k)\, s'_k$$

$$\sum_{i=1}^{n} \nu_{k,i}^2 = s_k'^2 \sum_{i=1}^{n} \mu_{k,i}^2 = s_k'^2 \,(n'_{k+1} - \varepsilon_{k+1}) \tag{11.17}$$

$$\sum_{i=1}^{n} \nu_{k,i}^2 = s'_{k+1} - \varepsilon_{k+1} s_k'^2$$

Multiplying (11.10) with s'_k gives

$$\nu_{k,1} = r[\mu_{k-1,i}^2 s_{k-1}'^2 + \varepsilon_{k,i} s_{k-1}'^2 + \eta_{k,i} s'_k]$$

Since

$$s'_k = \sum_{i=1}^{n} \nu_{k-1,i}^2 + \varepsilon_k s_{k-1}'^2 ,$$

we have

$$\nu_{k,i} = r\left[\nu_{k-1,i}^2 + (\varepsilon_{k,i} + \eta_{k,i}\varepsilon_k) s_{k-1}'^2 + \eta_{k,i} \sum_{i=1}^{n} \nu_{k-1,j}^2\right].$$

From (11.17)

$$s'_k = \frac{\sum_{i=1}^{n} \nu_{k,i}}{r(1+\eta_k)}$$

and hence

$$\nu_{k,i} = r\left[\nu^2_{k-1,i} + (\varepsilon_{k,i} + \eta_{k,i}\varepsilon_k)\frac{\left(\sum_{j=1}^{n}\nu_{k-1,j}\right)^2}{r^2(1+\eta_{k-1})^2} + \eta_{k,i}\sum_{j=1}^{n}\nu^2_{k-1,j}\right].$$

That is, if we define

$$\varepsilon^*_{k,i} = \frac{\varepsilon_{k,i} + \eta_{k,i}\varepsilon_k}{r^2(1+\eta_{k-1})^2} \tag{11.18}$$

then

$$\nu_{k,i} = r\left[\nu^2_{k-1,i} + \varepsilon^*_{k,i}\left(\sum_{j=1}^{n}\nu_{k-1,j}\right)^2 + \eta_{k,i}\sum_{j=1}^{n}\nu^2_{k-1,j}\right]. \tag{11.19}$$

We see too that

$$\frac{s'_k}{s_k} = \frac{1}{r}\,\frac{\sum_{i=1}^{n}\nu_{k,i}}{1+\eta_k}\,\frac{1}{r^{2^k-2}\sum_{i=1}^{n}\lambda_i^{2^k}}$$

$$= \frac{1}{r^{2^k-1}}\,\frac{\sum_{i=1}^{n}\nu_{k,i}}{\sum_{i=1}^{n}\lambda_i^{2^k}}\,\frac{1}{1+\eta_k}$$

and

$$\frac{\sum_{i=1}^{n}\nu_{k,i}}{r^{2^k-1}\sum_{i=1}^{n}\lambda_i^{2^k}} = \tag{11.20}$$

$$= \frac{\sum_{i=1}^{n}\nu_{k,i}}{r\sum_{i=1}^{n}\nu^2_{k-1,i}} \cdot \frac{\sum_{i=1}^{n}\nu^2_{k-1,i}}{r^2\sum_{i=1}^{n}\nu^4_{k-2,i}} \cdots \frac{\sum_{i=1}^{n}\nu^{2^s}_{k-s,i}}{r^{2^s}\sum_{i=1}^{n}\nu^{2^{s+1}}_{k-s-1,i}} \cdots \frac{\sum_{i=1}^{n}\nu^{2^{k-1}}_{1,i}}{r^{2^{k-1}}\sum_{i=1}^{n}\lambda_i^{2^k}}$$

If we let q_s be the general term of the above product and if we also let

$$m = k - s \quad , \quad N = 2^s$$

then

$$(11.21) \qquad q_s = \frac{1}{r^N} \frac{\sum_{i=1}^{n} \nu_{m,i}^N}{\sum_{i=1}^{n} \nu_{m-1,i}^{2N}}$$

In this notation

$$(11.22) \qquad \frac{s_k'}{s_k} = q_0 q_1 \cdots\cdots q_{k-1} \frac{1}{1+\eta_k}$$

12. Estimate of s_k' / s_k. We wish to obtain upper and lower bounds for s_k' / s_k. Since from (11.12)

$$(12.1) \qquad |\epsilon_{k,i}^*| \leq \alpha^* \text{ and } |\eta_{k,i}| \leq \beta$$

where

$$(12.1a) \qquad \alpha^* = \frac{\alpha(1+\beta)}{r^2(1-\beta)^2}$$

we have

$$(12.2) \qquad \frac{1}{r}\nu_{m,i} \leq \nu_{m-1,i}^2 + \alpha^* \left(\sum_{j=1}^{n} |\nu_{m-1,j}|\right)^2 + \sum_{j=1}^{n} \nu_{m-1,j}^2 = \frac{\nu_{m,i}}{r}$$

Hence

$$(12.3) \qquad q_s \leq \frac{\sum_{i=1}^{n} \left(\frac{\bar{\nu}_{m,i}}{r}\right)^N}{\sum |\nu_{m-1,i}|^{2N}} = \bar{q}_s$$

Put $$p_h = \frac{1}{n} \sum_{i=1}^{n} |\nu_{m-1,i}|^h$$

Then it follows that $\bar{q}_s$ is of the form

$$\sum_{(r_1 \cdots r_t)} a_{r_1 \cdots\cdots r_t} \frac{p_1^{r_1} \; p_2^{r_2} \cdots \; p_t^{r_t}}{P_{2N}}$$

where the coefficients a are non-negative constants (independent of the

$\nu_{n-1,j}$, but depending on n, s, $*$, and α), and where, because of the homogeneity of the expression (12.3) $r_1 + 2r_2 + .. + tr_t = 2N$

We now use the following facts:

(12.4) $$\frac{p_1^{r1} \quad p_2^{r}2 \cdot - \cdot p_t^{r}t}{p_{2N}} \leq 1.$$

This is proved below.

(12.5) If all $\nu_{m-1,j}$ are equal to each other, then the equality sign holds in (12.4)

This is evident from the definition p_h.

Hence $q_s \leq \sum_{(r_1..r_t)} a_{r_1....r_t} = \xi_s$ and ξ_s may be computed by setting all $\nu_{m-1,j}$ equal to c. We then have

$$\frac{\bar{\nu}_{m,i}}{r} = c^2 (1 + n\beta + n^2 \alpha *)$$

Hence with

(12.6) $$K = 1 + n\beta + n^2 \alpha *$$

we have

(12.7) $$\text{Max } q_s = K^N = K^{2^s}$$

Proof of inequality (12.4): Consider n non-negative numbers ξ_i.

Set $p_h = \frac{1}{n} \sum_{i=1}^{n} \xi_i^h$, $p_{h'} = \frac{1}{n} \sum_{i=1}^{n} \xi_i^h$,

$$p_{h+h'} = \frac{1}{n} \sum_{i=1}^{n} \xi_i^{h-h'}, \quad h \geq 0, \quad h' \geq 0$$

Then $P_{h+h'} - P_h \cdot P_{h'}$ may be written in the form

$$\frac{1}{n^2}\left(n\sum_{i=1}^{n}\xi_i^{h+h'}-\sum_{i=1}^{n}\xi_i^{h}\sum_{j=1}^{n}\xi_j^{h'}\right)=\frac{1}{n^2}\sum_{i<j}\left(\xi_i^{h+h'}+\xi_j^{h+h'}-\xi_i^{h}-\xi_j^{h}\xi_i^{h'}\right)$$

$$=\frac{1}{n^2}\sum_{i<j}\left(\xi_i^{h}-\xi_j^{h}\right)\left(\xi_i^{h'}-\xi_j^{h'}\right)$$

Since each product $(\xi_i^h - \xi_j^h)(\xi_i^{h'} - \xi_j^{h'})$ is non-negative, we find that $P_{h+h'} \geq P_h \cdot P_{h'}$ and by induction $P_{h_1} P_{h_2} \cdots P_{h_k} \leq P_{h_1 + h_2 + \cdots + h_k}$. If some of the h_i occur repeatedly we obtain the inequality (12.4). (Cf. Hardy, Littlewood, Polya, Inequalities, Cambridge 1934, p. 43).

We wish to obtain a lower bound for q_s. From the preceding work we see that

$$(12.8)\qquad q_s = \frac{\sum_{i=1}^{n}\left\{\nu_{m-1,i}^2 + \varepsilon^*_{m,i}\sum_{j=1}^{n}\nu_{m-1,j}^2 + \eta_{m,i}\sum_{j=1}^{n}\nu_{m-1,j}^2\right\}^N}{\sum_{j=1}^{n}\nu_{m-1,j}^{2N}}$$

We discuss q_s for fixed values $\nu_{m-1,j}$ as a function of the variables $x_i = \varepsilon^*_{m,i}$, $y_i = \eta_{m,i}$. Since the denominator does not depend on x_i or y_i we have

$$q_s = \text{const.} \sum_{i=1}^{n} f_i(x_i, y_i) = f(\vec{x}, \vec{y})$$

where $\vec{x}$ and $\vec{y}$ denote the vectors with the components x_i, y_i respectively. We see that $f(\vec{0},\vec{0}) = 1$. The f_i is of the form $(a_i + b_i x_i + c_i y_i)^N$, where N is an even number, or 1 (for s = 0). Since u^N is convex, that is

$$(12.9)\qquad u^N + v^N \geq 2\left(\frac{u+v}{2}\right)^N,$$

We have for every i

$$(12.10)\qquad f_i(x_i', y_i') + f_i(x_i'', y_i'') \geq 2 f_i\left(\frac{x_i' + x_i''}{2}, \frac{y_i' + y_i''}{2}\right)$$

Since all coefficients a_i, b_i, c_i in the expression for f_i are positive, cf. (12.8), it follows that $f_i(x_j, y_i) \geq f_i(0,0)$ if both x_j and y_i are

non-negative. Hence

$$f(\vec{x},\vec{y}) \geq f(\vec{0},\vec{0}) = 1 \tag{12.11}$$

if $\vec{x}$ and $\vec{y}$ have non-negative components. Denote by $\vec{e}$ the vector with components 1.....1. Because of the inequalities (12.1) both $\alpha * \vec{e}-\vec{x}$ and $\beta\vec{e} - \vec{y}$ have non-negative components.

By (12.10) and (12.11)

$$f(\vec{x},\vec{y}) + f(\alpha * \vec{e}-\vec{x}, \beta\vec{e}-\vec{y}) \geq 2 f\left(\frac{\alpha * \vec{e}}{2}, \frac{\beta\vec{e}}{2}\right) \geq 2 f(\vec{0},\vec{0}) = 2$$

and therefore

$$f(\vec{x},\vec{y}) \geq 2-f(\alpha * \vec{e}-\vec{x}, \beta\vec{e}-\vec{y}).$$

The absolute values of the components of $\alpha * \vec{e}-\vec{x}$, $\beta\vec{e}-\vec{y}$ are at most $\alpha *$ and β respectively. For this reason we may apply here our estimate of the maximum of q_s and we find

$$q_s = f(\vec{x},\vec{y}) \geq 2-K^{2^s} \tag{12.12}$$

We have now shown that

$$2-K^{2^s} \leq q_s \leq K^{2^s}$$

From this it follows with the use of (11.22) that

$$\frac{s'_k}{s_k} = \frac{q_0 q_1 \cdots\cdots q_{k-1}}{1+\eta_k} \leq \frac{K^{2^k-1}}{1-\beta} \tag{12.13}$$

A lower bound for $\frac{s'_k}{s_k}$ is given by

$$\frac{s'_k}{s_k} = \frac{q_0 q_1 \cdots\cdots q_{k-1}}{1+\eta_k} \geq \frac{2-K^{2^k-1}}{1+\beta} \tag{12.14}$$

where we assume explcitly that $2-K^{2^k-1} \geq 0$, and hence $2-K^{2^s-1} \geq 0$ for $0 \leq s \leq k-1$.

In order to see this we prove by induction that

$$q_0 q_1 \cdots\cdots q_{s-1} \geq 2-K^{2^s-1} .$$

It is clear that this is true when $s = 0$ by (12.12). Assuming it to be true for s, and remembering that $q_s \geq 2-K^{2^s}$, we have by multiplication

$$q_0 \cdots\cdots q_s \geq (2-K^{2^s-1})\,(2-K^{2^s}) = 2-K^{2^{s+1}-1} + 2(K^{2^s}-1)\,(K^{2^s-1}-1) .$$

The term $(K^{2^s}-1)\,(K^{2^s-1}-1)$ is positive and this completes the induction proof.

13. Computation of the Maximal Proper Value. After $n_1', \ldots, n_k'$ have been computed it remains to compute $n_1'^{1/2}\ n_2'^{1/4} \cdots\cdots n_k'^{1/2^k}$ in order to obtain an approximate expression for $r\lambda$ where λ is the largest proper value of A. Since this final computation will introduce new errors we must specify the manner in which it is to be made. It seems best to make this final computation by means of logarithms rather than by taking successive square roots, and in our estimates we assume that this final computation is done in this way.

From section 9,

$$(13.1) \qquad \left(\frac{r}{n}\right)^{2^{-k}} \leq \frac{r\lambda}{n_1^{1/2}\, n_2^{1/4} \cdots\cdots n_k^{2^{-k}}} \leq 1$$

and consequently

$$(13.2) \qquad \left|\log_b (r\lambda) - \log_b (n_1^{1/2}\ n_2^{1/4} \cdots\cdots n_k^{2^{-k}})\right| \leq 2^{-k} \log_b \frac{n}{r}$$

Here we use logarithms to the base b. The natural logarithm of y is denoted by log y so that if $c = \log b$ then $(\log_b y)\, c = \log y$. Let σ_k' be the computed value of $\log_b (n_1^{1/2}\ n_2^{1/4} \cdots\cdots n^{2^{-k}})$ and let τ_k be defined as follows

$$\tau_k = \left|\sigma_k' - \log_b (n_1^{1/2}\, n_2^{1/4} \cdots\cdots n_k^{2^{-k}})\right|$$

Then

(13.3) $$\left|\sigma'_k - \log_b r\right| \leq \tau_k + 2^{-k} \log_b \frac{n}{r}$$

and we now proceed to estimate τ_k and find conditions on k and δ such that the value of (13.3) is smaller than a prescribed error. We see from (11.3) and (11.6) that

(13.4) $$\frac{n'_k}{n_k} = \frac{s'_k}{s_k} \left(\frac{s'_{k-1}}{s_{k-1}}\right)^{-2}$$

Also from (12.13) and (12.14)

(13.5) $$\frac{(1-\beta)^2 (2-K^{2^k-1})}{(1+\beta) K^{2^k-2}} \leq \frac{n'_k}{n_k} \leq \frac{(1+\beta)^2 K^{2^k-1}}{(1-\beta)(2-K^{2^{k-1}-1})^2}$$

The numbers n'_k are such that

$$\frac{(r-\theta)^2}{n} \leq n'_k < 1$$

Clearly in taking the logarithms of these numbers we are forced to abandon the assumption that all numbers which occur are between -1 and +1. Since k will be a small number, we are actually concerned here with a very small number of operations by comparison with what has gone before.

Taking logarithms is assumed to introduce an error whose absolute value is bounded by ξ.

Denoting by log' the logarithm obtained by computation we have

$$\left|\log'_b x - \log_b x\right| \leq \xi$$

for any number x which is such that

$$\frac{(r-\theta)^2}{n} \leq x \leq 1$$

In computing

$$(1/2)\log_b n'_1 + (1/4) \log_b n'_2 + \ldots + \frac{1}{2^k} \log_b n'_k$$

two kinds of errors are involved (1) errors from taking logarithms (2) errors from division. Moreover, we must remember that there is an error coming from the fact that n_k' may differ from n_k. As we shall see, the first two kinds of errors are negligible compared to this last kind.

Let ϕ_s be the computed value of $\frac{1}{2^s} \log_b n_s$.

Then

$$\phi_s = \frac{1}{2^s} \log_b' n_s' + \xi_s = \frac{1}{2^s} (\log_b n_s' + \xi_s) + \xi_s$$

where ξ_s is the error arising from division and ξ_s is the error arising from taking logarithms. We also have

$$\phi_s = \frac{1}{2^s} \log_b n_s + \frac{1}{2^s} \log_b \frac{n_s'}{n_s} + \frac{1}{2^s} \xi_s + \xi_s . \tag{13.6}$$

Since

$$\sigma_k' = \sum_{s=1}^{k} \phi_s$$

where we assume that this addition introduces no error, we have

$$\left| \sigma_k' - \sum_{s=1}^{k} \frac{1}{2^s} \log_b n_s \right| \leq \sum_{s=1}^{k} \frac{1}{2^s} \log_b \frac{n_s'}{n_s} + \xi + k\delta . \tag{13.7}$$

Let

$$L_k = \sum_{s=1}^{k} \frac{1}{2^s} \log \frac{n_s'}{n_s} . \quad \text{Then } \tau_k \leq \frac{1}{c} L_k + \xi + k\delta \text{ and we}$$

proceed to estimate L_k.

From (13.5) and since $\dfrac{1+\beta}{(1-\beta)^2} > \dfrac{(1+\beta)^2}{1-\beta}$

$$L_k \leq \max\left\{ \sum_{s=1}^{k} \frac{1}{2^s} \left| \log \frac{K^{2^s-1}}{(2-K^{2^{s-1}-1})^2} \right| , \sum_{s=1}^{k} \frac{1}{2^s} \left| \log \frac{2-K^{2^s-1}}{K^{2^s-2}} \right| \right\} \tag{13.8}$$

$$+ \sum_{s=1}^{k} \frac{1}{2^s} \log \frac{1+\beta}{(1-\beta)^2}$$

Now let

(13.9) $$K = 1 + a$$

and we assume that

(13.10) $$2^k a \leq \frac{1}{2}$$

In section 12 we assumed that $2 - K^{2^k-1} \geq 0$ and we shall now show that the assumption (13.10) implies this earlier assumption. We see that (using (13.10))

$$K^{2^k-1} = (1+a)^{2^k-1} = e^{(2^k-1)\log(1+a)} \leq e^{(2^k-1)a} \leq e^{\frac{1}{2}} < 2$$

and this completes the proof of the fact that (13.10) implies that $2 - K^{2^k-1} \geq 0$.

As a first step toward obtaining an estimate for L notice that

$$K^t = (1+a)^t = e^{t \log(1+a)} < e^{ta}$$

and hence

(13.11) $$\log K^t < ta.$$

Therefore

(13.12) $$\frac{1}{2^s} \log K^{2^s-1} < \frac{(2^s-1)a}{2^s} < a$$

We write

$$L'_k = L'_{k1} + L'_{k2} + \log \frac{1+\beta}{(1-\beta)^2}$$

where

(13.13) $$L'_{k1} = \sum_{s=1}^{k} \frac{1}{2^s} \log K^{2^s-1}$$

and

(13.14) $$L'_{k2} = \sum_{s=1}^{k} \frac{1}{2^s} \log \frac{1}{(2-K^{2^{s-1}-1})^2}$$

$$= \sum_{s=1}^{k} \frac{1}{2^{s-1}} \log \frac{1}{1-(K^{2^{s-1}-1}-1)}$$

Thus L_k' is the sum on the right of (13.8) using the first of the two terms in the max, and it will be necessary to obtain a bound for L_k'.

From (13.12) we have

$$L_{k1}' \leq a(k-1+2^{-k}) \tag{13.15}$$

Notice now that

$$L_{k2}' = \sum_{s=1}^{k-1} \frac{1}{2^s} \log \frac{1}{1-(K^{2^s-1}-1)} \tag{13.16}$$

We will make use of certain inequalities for positive numbers.

$$e^x-1 < x + \frac{x^2}{2}(1 + x/3 + x^2/9 + \ldots.) = x + \frac{x^2}{2}\frac{1}{1-x/3} < x + \frac{\gamma_1 x^2}{2} \tag{13.17}$$

where $\gamma_1 < \frac{1}{1-x/3}$ if $x_.$ is the largest value of x occurring and if $x < \frac{1}{2}$ then $\gamma_1 < 6/5$.

Another inequality of use is

$$\log \frac{1}{1-y} = y + \frac{y^2}{2} + \ldots. < y + \frac{y^2}{2} + \frac{y^3}{3}\frac{1}{1-y} < y + \frac{y^2}{2} + \gamma_2 \frac{y^3}{3} \tag{13.18}$$

where γ_2 may be chosen as $\frac{1}{1-y_.}$ if $y_.$ is the largest value of y occurring.

Let

$$y_s = K^{2^s-1}-1 \quad , \quad x_s = (2^s-1)a. \tag{13.19}$$

Then

$$y_s = e^{(2^s-1)\log(1+a)} - 1 < e^{(2^s-1)a} - 1 = e^{x_s}-1 < x_s\left(1 + \frac{\gamma_1}{2} x_s\right) . \tag{13.20}$$

Since $x_s < \frac{1}{2}$, $y_s < \frac{13}{20}$, and γ_2 may be chosen as $\frac{20}{7}$. Using (13.18), (13.19) and (13.20)

$$\frac{1}{2^s} \log \frac{1}{1-y_s} < \frac{1}{2^s} y_s \left(1 + \frac{1}{2} y_s + \frac{20}{21} y_s^2\right) \tag{13.21}$$

$$< \frac{1}{2^s}\left(x_s\left(1+\frac{3}{5}x_s\right)\right)\left(1+\frac{1}{2}y_s+\frac{20}{21}y_s^2\right)$$

$$< \left(1-\frac{1}{2^s}\right) a \left(1+\frac{3}{5}x_s\right)\left(1+\frac{1}{2}y_s+\frac{20}{21}y_s^2\right)$$

Making use of (13.19) and (13.20) we change this into an expression in powers of a. We also use the inequality

$$\sum_{s=1}^{k-1} x_s^m < a^m \sum_{s=1}^{k-1} 2^{ms} < \frac{(2^k a)^m}{2^m-1}.$$

Omitting the details we obtain, (also using $2^k a < \frac{1}{2}$)

(13.22) $$L_{k2} < a\left[(k-2+2^{-(k-1)}) + 2^k a\,(1.1 + (2^k a))\right]$$

Together with (13.15) this gives (since $\log \frac{(1+\beta)}{(1-\beta)} 2 \leq 4\beta$ if $\beta < \frac{1}{2}$, and this is the case by our previous requirements).

(13.23) $$L_k' < a\left[2k-3 + 3.2^{-k} + 2^k a\,(1.1 + (2^k a))\right] + 4\beta$$

Letting L_k'' be a bound for the sum on the right of (13.8), using the second of the two terms in the max and also letting

$$L_{k1}'' = \sum_{s=1}^{k} \log K^{2^s-2}$$

$$L_{k2}'' = \sum_{s=1}^{k} \frac{1}{2^s} \log \frac{1}{1-(K^{2^s-1}-1)}$$

It follows that

$$L_k'' \leq L_{k1}'' + L_{k2}'' + 4\beta$$

and we see that

$$L_{k1}'' \leq a(k-2+2^{-(k-1)})$$

The same procedure as before applies to L_{k2}'', the only difference being that sums run from 1 to k instead of from 1 to k-1. We obtain

(13.24) $$L_{k2}'' < a\left[(k-1+2^{-k}) + 2^k a\,(2.2 + 4\,(2^k a))\right]$$

and therefore

$$L_k'' < a[2k-3 + 3.2^{-k} + 2^k a\ (2.2 + 4(2^k a))] + 4\beta \tag{13.25}$$

The estimate for L_k'' is larger than that for L_k' so that L_k is at most equal to L_k'' and as a result

$$L_k < a\ [\ 2k-3\ + 3.2^{-k} + 2^k a\ (2.2 + 4\ (2^k a))\ ] + 4\beta \tag{13.26}$$

Since $a = n^2 \alpha^* + n\beta$, and since α^* itself contains a factor nz it is clear that ξ, and $k\ \delta$ are negligible compared with $\frac{1}{c} L_k$. Therefore τ_k is essentially given by $\frac{1}{c} L_k$.

Returning to (13.3) and inserting the values we have obtained

$$|\sigma_k' - \log_b r\lambda| \leq 2^{-k} \log_b \frac{n}{r} + \frac{1}{c} L_k + \xi + k\delta\ . \tag{13.27}$$

In computing λ, two additional errors are introduced, the first in computing $\log_b r$ and in taking the exponential. As a matter of fact, the numerical examples below show r can be chosen so near to one that within the error of the method λ can be replaced by λr. These are negligible compared to $\frac{1}{c} L_k$. Suppose we want an accuracy d by which we mean if λ' is the computed value

$$(1-d)\ \lambda' < \lambda < (1+d)\ \lambda'\ .$$

Then essentially $\frac{d}{c}$ is the maximum error to be permitted in the computed value of the logarithm. This accuracy can be obtained by requiring

$$2^{-k} \log_b \frac{n}{r} < d/2c$$

which determines the number of steps k, and $L_k < \frac{d}{2}$ which determines the number of digits to be carried. Writing these more explicitly, we have

$$\frac{1}{2^k} \log_b \frac{n}{r} < \frac{d}{2c} \tag{13.28}$$

$$4\beta + a\ [2k-3\ + 3.2^{-k} + 2^k a\ (2.2 + 4\ (2^k a))] < d/2$$

Summarizing the definition of the quantities involved and the requirements

we have made

$$\alpha^* = \frac{\alpha(1+\beta)}{r^2(1-\beta)^2}, \quad .9 < r \leq 1-4n^3\delta, \quad \alpha = nz\delta, \quad \beta = \frac{2n^2\delta}{r}$$

(13.29)

$$a = n^2\alpha^* + n\beta, \quad n \geq 3, \quad n^3\delta \leq 1/50, \quad 2^k a \leq \tfrac{1}{2}.$$

It is always best to choose r as large as possible and in the estimates of (13.28) r can safely be regarded as one.

Without any statistical estimate, $z = n$. Making a statistical estimate we proceed as follows. The number z is to be the error in a single element of a product matrix. It is the sum of n errors which we assume are independent and uniformly distributed between $-\delta$ and $+\delta$. For a single error, the mean is zero and the dispersion is $\frac{\delta^2}{3}$. Consequently for sufficiently large n, we will probably obtain an approximately normal distribution

(13.30) $$\frac{1}{\sigma}\sqrt{\frac{1}{2\pi}}\, e^{-\frac{\xi^2}{2\sigma^2}}$$

where $\sigma^2 = \frac{n\delta^2}{3}$. With a probability > .99 the absolute value of an element of the error matrix X is smaller than $2.33\,\sigma = 1.35\sqrt{n}\,\delta$ so that z may be taken to be $1.35\sqrt{n}$.

It should be noted that we base our estimate for the bound on the highest bound attainable for elements with a given maximum absolute value. If we analyze the statistical distribution of the bound of a matrix whose elements are independent are all distributed according to (13.30); we may gain another factor of the order $\sqrt{n}$. However we do not attempt this at this time.

14. Remarks and Examples. In this section we give some estimates on the values of k, and the number of digits necessary in some typical cases. If we take $\alpha = n^2\delta$ we get estimates not using the statistical

distribution or errors in multiplication. If we wish to take account of the latter we must take $\alpha = 1.35\ n^{3/2}\delta$.

In the following examples k is the number of steps necessary to attain an accuracy d, p_1 is the number of decimal digits which must be carried if we make no statistical estimate of the round off error, and p_2 is the number of steps with such a statistical estimate.

Example I, n = 10 , log n = 2.3

d	k	p_1	p_2
.1	6	7-	6
.01	9	8	7
.001	13-	9	9-

Example II, n = 20 , log n = 3.0

d	k	p_1	p_2
.1	6	8-	7
.01	10-	9-	8+
.001	13	10-	9+
.0001	16	11	11-

Example III, n = 30 , log n = 3.4

d	k	p_1	p_2
.1	7-	8	8
.01	10	9+	9
.001	13	11-	10
.0001	16+	12-	11

Example IV, n = 50 , log n = 3.9

d	k	p_1	p_2
.1	7-	9	8+
.01	10	10+	10-
.001	13	11+	11-
.0001	17-	13-	12
.00001	20-	14-	13

Example V, n = 100 , log n = 4.6

d	k	p_1	p_2
.1	7	11	10
.01	10	12	11
.001	14-	13	12
.0001	17	14	13
.00001	20	15	14

Example VI, n = 1000 , log n = 6.9

d	k	p_1	p_2
.1	8-	15	13
.01	11	16	14
.001	14	17	15+
.0001	18-	18	17-
.000001	24	20	19-

Although at present it is not practical to deal with matrices of order 1000, the last example is inserted to show the dependence of p_1 and p_2 on n.

We observe that the number of digits required is fairly moderate even in case n = 1000, and that k, the number of steps required increases very slowly with n. These two facts imply that the amount of labor involved is roughly proportional to n^3 and not to some higher power of n as might conceivably be the case if either k or p increased rapidly. For large values of n the amount of labor is very great, but there is no doubt that high speed electronic machines now being built will tremendously increase the size of n available.

15. Computation of the Smallest Proper Value. In this section it will be shown how the above discussion can be modified to apply to the computation of the minimum proper value.

Let A be a matrix of the kind we have been considering, where r is subject to the limitations stated previously. We assume that λ , the maximum proper value of A, is known with a relative accuracy d, that is that λ', the computed value, and the actual value satisfy

(15.1) $$\lambda' (1-d) < \lambda < \lambda' (1+d) .$$

The computation of the minimum proper value of A can be reduced to the computation of the maximum proper value of $\rho \cdot 1 - A$ where ρ is so chosen that $\rho \cdot 1 - A$ is positive definite. Since A has a bound smaller than one, $\rho = 1$ is always a possible choice. Our discussion will show that it is advantageous to choose ρ as small as possible, and for this reason ρ is chosen as follows:

(15.2) $$\rho = \min [1-2n\delta , \lambda' (1+d)] .$$

With this choice the matrix

(15.3) $$H = \rho \cdot 1 - A$$

is positive definite and

(15.4) $$t(H) = n\rho - t(A)$$

Also the largest proper value of H is less than one (10.11b). We wish to form the matrix

(15.5) $$G = \frac{rH}{t(H)}$$

but by computation we obtain

(15.6) $$G' = \frac{rH}{t(H)} + X'$$

where we first compute $\frac{H}{t(H)}$ and then multiply by r. Since H is positive definite, every matrix element of H has absolute value at most t(H) so that every division involved will have an error at most equal to δ'. The quantity δ' is equal to $\frac{1}{2}10^{-p'}$ or $\frac{1}{2}2^{-p'}$ if p' is the number of decimal or binary digits. In general it is advisable to have p' larger than the p used in computing the maximal proper value. Since multiplication by r introduces errors whose absolute values have the same bound, each element of X' has absolute value at most 2 δ' and therefore

(15.7) $$|X'| \leq 2n\ \delta' , \quad |t(X')| \leq 2n\ \delta' .$$

From (15.6) we see that

(15.8) $$r-\theta' < t(G') < r + \theta'$$

where

(15.9) $$\theta' = 2n^2\ \delta'$$

so that (10.13) is satisfied. Let λ_1 be the maximum proper value of G. Then if μ is the minimum proper value of A,

(15.10) $$\lambda_1 = \frac{\rho - \mu}{n\rho - r}$$

If λ_2 is the maximum proper value of G', we have $\lambda_2 = \lambda_1 + \xi$

where

$$|\xi| \leq |x'| \leq 2n\delta'$$

Let us assume now that λ_2' is the computed maximal proper value of G' and that is has been computed with an accuracy d' so that

$$(15.12) \qquad (1-d')\,\lambda_2' \leq \lambda_2 \leq (1+d')\,\lambda_2' ,$$

and

$$(15.13) \qquad (1-d')\,\lambda_2' \leq \frac{\rho-\mu}{n\rho-r} + \xi \leq (1+d')\,\lambda_2' .$$

From this follows

$$(15.14) \qquad \begin{aligned} &\rho - (n\rho - r)(1+d')\lambda_2' + (n\rho - r)\xi \leq \mu \\ &\leq \rho - (n\rho - r)(1-d')\lambda_2' + (n\rho - r)\xi . \end{aligned}$$

Since from (15.10)

$$(15.15) \qquad \mu = \rho - (n\rho - r)\,\lambda_1$$

it is natural to use this for finding the computed value of μ from the computed value of λ_1. Of course making this computation itself involves an error so that if μ' is the computed value of μ we have

$$(15.16) \qquad \mu' = \rho - (n\rho - r)\,\lambda_2' + \xi$$

where

$$(15.17) \qquad \xi \leq n\delta'$$

Then, substituting in (15.14),

$$\rho - (\rho - \mu' + \xi)(1+d') + (n\rho - r)\xi \leq \mu$$

$$\leq \rho - (\rho - \mu' + \xi)(1-d') + (n\rho - r)\xi$$

or

$$\mu'(1+d') + [(n\rho - r)\xi - \rho d' - (1+d')\xi] \leq \mu$$

$$\leq \mu'(1-d') + [(n\rho - r)\xi + \rho d' - (1-d')\xi]$$

Rearranging and dividing by μ'

$$1 + d' + [(-d' + n\xi)\ \frac{\rho}{\mu'} + \frac{-r\xi - (1+d')\xi}{\mu'}] \leq \frac{\mu}{\mu'}$$

(15.18)

$$\leq 1 - d' + [(d' + n\xi)\ \frac{\rho}{\mu'} + \frac{-r\xi - (1-d')\xi}{\mu'}]$$

Remembering (15.11) and (15.17) we may write

$$1 - (d' + 2n^2\sigma')\ \frac{\rho}{\mu'} + d' - \frac{r\ 2n\sigma' + (1+d')\ n\sigma'}{\mu'} \leq \frac{\mu}{\mu'}$$

$$\leq 1 + (d' + 2n^2\sigma')\ \frac{\rho}{\mu'} - d' + \frac{r\ 2n\sigma' + (1+d')\ n\sigma'}{\mu'}$$

Since $r \leq 1$ and $d' < 1$ we may also write

$$1 - (d' + 2n^2\sigma')\ \frac{\rho}{\mu'} + d' - 4\ \frac{n\sigma'}{\mu'} \leq \frac{\mu}{\mu'}$$

(15.19)

$$\leq 1 + (d' + 2n^2\sigma')\ \frac{\rho}{\mu'} - d' + 4\ \frac{n\sigma'}{\mu'}\ .$$

In general it may be expected that the dominating term in this expression for the accuracy of μ' will be $\frac{\rho}{\mu'}$, and it is for this reason that we suggested the desirability of keeping as small as possible. To sum up, if we let

$$d^* = (d' + 2n^2\sigma')\ \frac{\rho}{\mu'} + 4\ \frac{n\sigma'}{\mu'} - d' \qquad (15.20)$$

we have

$$(1-d^*)\mu' \leq \mu \leq (1+d^*)\mu'\ . \qquad (15.21)$$

From a statistical analysis which will be discussed in a later report, we may draw some conclusions about the probable size of $\frac{\rho}{\mu'}$. In this analysis it is assumed that M is a matrix about which we know only that its elements have the same normal distribution. Under this assumption it is shown that if $A = M^*M$, then at least for large n, the probable ratio of the largest to the smallest proper value is about n^2. This ratio can also be used in the case when the matrix is not given at random but arises from the attempt to solve

approximately an elliptic partial differential equation by replacing it by n linear equations. In view of (15.20) it is therefore reasonable to expect that d' must be in general of the order $1/n^2$.

By this computation we shall find either that the smallest proper value is safely away from zero and that we may proceed to find the inverse or that the smallest proper value is not greater than a certain positive number ϵ so that within the given accuracy we can not decide whether or not the matrix has an inverse. In this latter case it is impossible to proceed further.

If we have decided that we wish μ to be greater than ϵ, we see from (15.19) that a sufficient condition for concluding this to be the case is

$$\mu' (1+d') > \epsilon + \rho (d' + 2n^2 \delta') + 4n \delta' \tag{15.22}$$

If this inequality is satisfied we are certain that A is positive definite and only in this case would we proceed to find the inverse.

It may be pointed out that it is not necessary to know the highest proper value with great accuracy because changing ρ by a few percent does not affect (15.22) appreciably. On the other hand it is clear that considerably greater accuracy is required in the computation of μ .

16. Theoretical Determination of the Inverse Matrix. One method which has been suggested for finding the inverse of a matrix (Hotelling, loc. cit.) is an iterative procedure. If A is a positive definite symmetric matrix of bound less than one, this iterative scheme may be defined in the following way:

$$\begin{aligned} F_0 &= 1 \\ F_{k+1} &= F_k (2-AF_k). \end{aligned} \tag{16.1}$$

In order to discuss convergence and to show that F_k approaches A^{-1} let

$$C_o = A$$

$$C_k = AF_k \quad .$$

Then

$$C_{k+1} = C_k\,(2-C_k)$$

and hence

$$1-C_{k+1} = (1-C_k)^2$$

so that

$$1-C_k = (1-A)^{2^k}$$

Taking bounds we see that

$$\left|1-C_k\right| = \left|1-A\right|^{2^k} = (1-\mu)^{2^k}$$

where μ is the smallest proper value of A. If $\mu > 0$, then $1-C_k$ must converge to 0 and the rate of convergence is given by the above equation. From the definitions

$$A^{-1}-F_k = A^{-1}-A^{-1}\,C_k = A^{-1}\,(1-C_k) = A^{-1}\,(1-A)^{2^k}$$

Hence

$$\left|A^{-1}-F_k\right| = \frac{(1-\mu)^{2^k}}{\mu}$$

and we see that F_k approaches A^{-1} with a rate of convergence given by this equation.

It should be noticed that for the definition of this procedure it is unnecessary to know the largest and smallest proper values. It can also be shown (Hotelling, loc. cit.) that estimates of the error involved may sometimes be made without knowledge of the minimum proper value. We show that when C_k approaches 1, F_k approaches A^{-1} and for this purpose we proceed as follows:

(16.1a)
$$A^{-1} = F_k\,C_k^{-1} = F_k\,[1-(1-C_k)]^{-1}$$

Hence

$$A^{-1}-F_k = F_k\left\{[1-(1-C_k)]^{-1}-1\right\} = F_k\,(1-C_k)\,[1-(1-C_k)]^{-1}$$

At any stage of the iterative process this formula can be used to estimate $A^{-1}-F_k$. Taking bounds

$$|A^{-1}-F_k| \leq |F_k| \cdot \frac{|1-C_k|}{1-|1-C_k|}$$

and replacing bounds on the right by norms

$$|A^{-1}-F_k| \leq N(F_k)\ \frac{N(1-C_k)}{1-N(1-C_k)}\ .$$

For this inequality we assume that not only the bound but also the norm of $1-C_k$ is less than 1. This will certainly be true for sufficiently large k if C_k converges to 1 and it is advantageous to use norms because they are readily computed.

If the minimum proper value is unknown in advance then it will be unknown whether or not C_k converges to one. In case it converges it will also be unknown how rapidly it converges and it will be unknown in advance how big F_k's will become. Therefore it will not be known in advance how many digits must be used.

If the minimum proper value is known in advance then it will be known whether or not A^{-1} exists. Furthermore a knowledge of the minimum proper value can be used to speed the convergence as we shall point out. This can be done by modifying the iterative process.

In considering the preceding method in general terms we notice a) that C_{k+1} is a quadratic function of C_k b) that the maximum proper value of C_k is at most one. The question arises as to whether there might not be a quadratic function satisfying these conditions and at the same time making the minimum proper value increase more rapidly. This quadratic function must be such that it also defines F_{k+1} in terms of F_k which means that it should be of the form

(16.2) $$c_k (\alpha_k - \beta_k c_k) \ .$$

If we consider the function

(16.3) $$y = x(\alpha - \beta x) = f(x)$$

we must require

(16.4) $$\text{if } 0 \leq x \leq 1, \text{ then } x \leq y \leq 1 \ ,$$

if nothing is known about the smallest proper value. However, if the minimal proper value is equal to $\mu < 1$, then we must require

(16.5) $$\text{if } \mu \leq x \leq 1 \ , \text{ then } \mu < y \leq 1 \ .$$

In the first case, that is the case where nothing is known about the smallest proper value, we see that f(x) must be monotonically increasing from 0 to 1, as x increases from 0 to 1. Among functions of this kind the function $f(x) = x\ (2-x)$ is the best choice. It increases the minimal proper value most rapidly, that is for this function, at every point in the open interval 0 to 1, f(x)-x is greater than the corresponding increase for any other function compatible with the conditions, as is easily verified. This coincides with the choice (16.1) of the iterative process already described.

In the second case, that is the case where μ is known, a more advantageous choice can be made as we will now indicate. The best choice in this case would be the one in which the minimum of f(x) in the interval $\mu \leq x \leq 1$ is highest. By means of an elementary discussion it can be shown that this function must be symmetric about the midpoint of the interval $(\mu, 1)$ where it reaches its maximum value one. From this it follows that f is of the form

(16.6) $$f(x) = \frac{4x}{1+\mu} \left(1 - \frac{x}{1+\mu} \right) \ .$$

Then the minimum value of f in the interval $\mu \leq x \leq 1$ is

$$f(\mu) = f(1) = \frac{4\mu}{(1+\mu)^2} \tag{16.7}$$

We also notice that

$$1-f(\mu) = \left(\frac{1-\mu}{1+\mu}\right)^2 \tag{16.8}$$

Using the function (16.6) the iterative procedure is defined as follows:

$$\mu_{k+1} = \frac{4\mu_k}{1+\mu_k^2}$$

$$\mu_o = \mu \tag{16.9}$$

$$F_{k+1} = \frac{4F_k}{1+\mu_k}\left(1 - \frac{AF_k}{1+\mu_k}\right)$$

$$F_o = 1$$

If $C_k = AF_k$, we have

$$C_o = A \tag{16.10}$$

$$C_{k+1} = \frac{4C_k}{1+\mu_k}\left(1 - \frac{C_k}{1+\mu_k}\right)$$

and hence

$$1 - C_{k+1} = \left(1 - \frac{2C_k}{1+\mu_k}\right)^2 \tag{16.11}$$

With this method the smallest proper value of C_k is μ_k and therefore

$$\left|1 - C_{k+1}\right| = 1 - \mu_{k+1} = \left(\frac{1-\mu_k}{1+\mu_k}\right)^2 \tag{16.12}$$

When μ_k is small we see from (16.9) that one of the iterative steps multiplies it approximately by 4 and when μ_k is near 1 we see from

(16.12) that, approximately, $1-\mu_{k+1} = 1/4\ (1-\mu_k)^2$. In the previous case when μ_k was near zero it was approximately multiplied by two in each step, and when μ_k was near one, $1-\mu_{k+1} = (1-\mu_k)^2$.

Consequently we see that in the second method μ_k increases more rapidly than it did in the first method. As an illustration of the comparative speeds we give here two numerical examples. In both cases we assume we want an accuracy of 1 per thousand in the inverse matrix which implies for the two cases different bounds for $1-C_k$.

μ	$\lvert 1-C_k \rvert$	Number of Steps for Method I	Number of Steps for Method II
.01	10^{-5}	11	6
.0001	10^{-7}	18	10

Although the second method looks more favorable it must not be forgotten that in order to obtain μ several matrix multiplications are necessary. It should, however, also be remembered that each step in the inversion procedure requires two matrix multiplications, as compared to only one multiplication in every step for the procedure for obtaining an extreme proper value. Also in the later stages of the inversion process it is necessary to carry more digits. In the examples above there is a saving of 10 matrix multiplications in the first case and 16 matrix multiplications in the second case when method II is used instead of method I. Below we shall point out that a knowledge of the maximum proper value saves considerable labor either by method I or method II, and therefore in either case it is certainly advisable to compute the maximum proper value. When we compute the minimum proper value and use method II, there may be a small percentage of extra labor as compared with method I, but we must remember that it gives us more information.

A knowledge of the maximum proper value saves labor in the computation of the inverse, because with this knowledge we may normalize the matrix so that the maximum proper value is about one. This increases the minimum proper value by a corresponding amount and, accordingly, fewer steps are required, since we have seen in both methods that the rate of convergence depends on the size of the minimum proper value.

17. Numerical Computation of the Inverse. In applying the theoretical discussion of the preceding section, certain slight modifications are necessary in order to take into account the occurrence of round off errors. We now proceed to rigorously describe method II, thus suitably modified for computation. One important consideration is that round off errors must be prevented from accumulating so that the bound of the computed C_k becomes greater than one. We must also keep in mind that while we cannot expect to know the minimum of C_k precisely, we must be sure that some definite lower bound for this minimum increases as rapidly as possible toward one.

We also point out that we no longer attempt to devise a process in which all numbers occuring are between -1 and +1 , although it would be possible to do so by fairly simple changes in the following procedure. We assume that a fixed number p of digits occur at the right of the decimal point.

We assume that A is a symmetric matrix. We also assume that the maximum and minimum proper values have been determined within certain specific limits of error. In what follows it is only necessary to know numbers μ^* and λ^* such that if μ and λ are the smallest and largest proper values of A then

$$0 < \mu^* \leq \mu \ , \qquad \lambda \leq \lambda^* \leq 1 .$$

For the reasons mentioned in the preceding section we wish to normalize A so that the maximum proper value is less than one but as near one as possible. Choose a number λ_o such that if δ'' is the round off error in a single multiplication or division, then

$$\lambda_o \geq \frac{\lambda^*}{1 - n\,\delta''} \tag{17.1}$$

We next divide A by λ_o. Remembering round off errors this leads to a matrix

$$A_o = \frac{A}{\lambda_o} + X \tag{17.2}$$

where

$$|X| \leq n\,\delta'' \tag{17.3}$$

Then

$$|A_o| \leq \frac{\lambda^*}{\lambda_o} + n\,\delta'' \leq 1$$

and for the minimum proper value we obtain the following estimate

$$\text{Min } A_o \geq \frac{\mu'}{\lambda_o} - n\,\delta'' = \varepsilon_o \,. \tag{17.4}$$

We assume explicitly that $\varepsilon_o > 0$.

We proceed to describe in detail the method for computing the inverse of A_o. We change the procedure described in (16.9) by inserting a safety factor α_k which is slightly less than one and by replacing μ_k by a certain lower bound ε_k for μ_k. The relation between ε_{k+1} and ε_k is not the same as that given in (16.9) for the relation between μ_{k+1} and μ_k but must be changed accordingly. The procedure thus becomes

$$F_o = 1$$

$$F_{k+1} = \frac{4\alpha_k F_k}{1 + \varepsilon_k}\left(1 - \frac{A_o F_k}{1 + \varepsilon_k}\right) \tag{17.5}$$

with constants α_k and ε_k to be determined later. The actual computed matrices are denoted by F'_k and we have

$$F'_o = F_o = 1$$

(17.6)

$$F'_{k+1} = \frac{4\alpha_k F'_k}{1+\varepsilon_k}\left(1 - \frac{A_o F'_k}{1+\varepsilon_k}\right) + R_k$$

where R_k is the error matrix. We have also, if $C'_k = A_o F'_k$, that

$$C'_{k+1} = \frac{4\alpha_k C'_k}{1+\varepsilon_k}\cdot\left(1 - \frac{C'_k}{1+\varepsilon_k}\right) + AR_k \,. \tag{17.7}$$

The numbers α_k and ε_k will be so chosen that

$$0 < \alpha_k < 1$$

(17.8)

$$0 < \varepsilon_k < 1 \,.$$

We wish to make this choice in such a way that

$$|C'_k| \leq 1$$

(17.9)

$$\text{Min } C'_k \geq \varepsilon_k \,.$$

From the definition of C'_k, we see that

$$|F'_k| \leq |A_o^{-1}| \cdot |C'_k| \,. \tag{17.10}$$

Hence if we can control $|C'_k|$ we will also be able to control $|F'_k|$.

Since δ'' is the round off error in a single multiplication

$$R_k \leq n\left\{\frac{4\alpha_k}{1+\varepsilon_k}\left[n(n+1)\delta''|F'_k| + n\delta''\right] + \delta''\right\} \tag{17.11}$$

so that except for negligible terms

$$(17.12) \qquad |R_k| \leq \frac{4\alpha_k}{1+\varepsilon_k} n^3 |F_k'| \delta'' .$$

If we wish to estimate errors statistically, this absolute bound could be lowered, essentially by replacing n^3 by n^2.

Let

$$(17.13) \qquad \rho_k \geq |R_k|$$

and recall that

$$\varepsilon_o \leq \min A_o = \min C_o' .$$

We also assume $\rho_k \leq \varepsilon_o$. It is clear that (17.9) is satisfied when $k = 0$, and we assume now that we have reached the k^{th} level with (17.9) satisfied. We shall prove it at level k+1 by induction.

If $\gamma'_{k,i}$ are the proper values of C_k', in a suitable order, then

$$(17.14) \qquad \gamma'_{k+1,i} = \frac{4\alpha_k \gamma'_{k,i}}{1+\varepsilon_k}\left(1 - \frac{\gamma'_{k,i}}{1+\varepsilon_k}\right) + \eta_{k,i}\rho_k$$

where

$$|\eta_{ki}| \leq 1 .$$

Now consider the funtion

$$(17.15) \qquad f(x) = \frac{4\alpha_k x}{1+\varepsilon_k}\left(1 - \frac{x}{1+\varepsilon_k}\right).$$

We see that

$$(17.16) \qquad f(o) = 0 \ , \ f(\varepsilon_k) = f(1) = \frac{4\alpha_k \varepsilon_k}{(1+\varepsilon_k)^2}$$

and also that the maximum occurs at $x = \frac{1+\varepsilon_k}{2}$ and is

$$(17.17) \qquad f_{max} = \alpha_k \text{ when } x = \frac{1+\varepsilon_k}{2}$$

Clearly for $\varepsilon_k \leq x \leq 1$ we have

(17.18) $$f(x) \geq \frac{4\alpha_k \varepsilon_k}{(1+\varepsilon_k)^2} .$$

By the hypothesis of the induction

$$\varepsilon_k \leq \gamma'_{k,i} \leq 1$$

and this implies

(17.19) $$\frac{4\alpha_k \varepsilon_k}{(1+\varepsilon_k)^2} - \rho_k \leq \gamma'_{k+1,i} \leq \alpha_k + \rho_k .$$

We now choose

(17.20) $$\alpha_k = 1 - \rho_k$$

and

(17.21) $$\varepsilon_k = \frac{4\alpha_{k-1}\varepsilon_{k-1}}{(1+\varepsilon_{k-1})^2} - \rho_{k-1}.$$

Then

(17.22) $$|C'_{k+1}| \leq 1 , \text{ and } \min C'_k \geq \varepsilon_k .$$

From this it follows from (17.10) that

(17.23) $$|F'_k| \leq |A_o^{-1}| .$$

It may be desirable in some cases to let ρ_k be variable but it may also be desirable in many cases to choose ρ_k as constant and in fact as follows:

(17.24) $$\rho = \rho_k = 4n^3(1/\varepsilon_o)\delta^n .$$

We see that

$$\varepsilon_{k+1} \geq \frac{4\varepsilon_k}{(1+\varepsilon_k)^2}(1-\rho) - \rho \geq \frac{4\varepsilon_k}{(1+\varepsilon_k)^2} - 2\rho$$

(17.25) $$1 - \varepsilon_{k+1} \leq \left(\frac{1-\varepsilon_k}{1+\varepsilon_k}\right)^2 + 2\rho .$$

We shall now see that in the process we have described, ε_k gets

near 1. Of course the existence of round off errors makes it impossible to get nearer to 1 than a certain very small positive quantity. We shall first prove that some ε_k must be greater than 1/10. If on the contrary ε_k were never greater than 1/10 the following formula would always be true

$$\varepsilon_{k+1} \geq \frac{4}{(1.1)^2}\varepsilon_k - 2\rho .$$

We can then see by induction that

$$\varepsilon_k \geq \left(\frac{4}{(1.1)^2}\right)^k \left(\varepsilon_0 - \frac{2\rho}{\frac{4}{(1.1)^2}-1}\right) + \frac{2\rho}{\frac{4}{(1.1)^2}-1}$$

(17.26)

$$\geq (3.3)^k \left(\varepsilon_0 - \frac{2}{2.3}\rho\right) .$$

Since it has been assumed that $\rho < \varepsilon_0$, this formula shows that for sufficiently large k, ε_k must get large, in particular it must get larger than 1/10. Hence we have been led to a contradiction by assuming that ε_k is always less than or equal to 1/10 , and we have therefore established that some $\varepsilon_k > 1/10$.

We now assume further that

(17.27) $$2\rho \leq .01$$

which would appear to be a very modest restriction. From (17.25) we see that once any $\varepsilon_{k'}$ is > .1, then $\varepsilon_{k'+1} > .3$, $\varepsilon_{k'+2} > .7$ and $\varepsilon_{k'+3} > .95$.

Beginning at this point we have the following inequality for all subsequent k

$$1 - \varepsilon_{k+1} \leq \frac{1}{3.8}(1-\varepsilon_k)^2 + 2\rho .$$

This shows that ε_k is monotonic increasing as long as

$$\rho \leq \left(1 - \frac{1-\varepsilon_k}{3.8}\right) \qquad \frac{1-\varepsilon_k}{2}$$

that is (a fortiori) as long as

$$\rho \leq (1 - \frac{1-.95}{3.8}) \frac{1-\varepsilon_k}{2} = \frac{1-\varepsilon_k}{2.03}, \text{ or } \varepsilon_k \leq 1 - 2.03\rho$$

and this gives us a good estimate of how near 1 we may approach.

Let us return to the step by step process we described above. If $2\rho \leq 10^{-4}$, then $\varepsilon_{k'+4} > 1-.0026$ and if $2\rho < 10^{-(4+r)}$ then at a fifth step

$$\varepsilon_{k'+5} > \begin{cases} 1 - 10^{-5} \\ 1 - 10^{-(4+r)} \end{cases}$$

whichever is smaller. If ρ is sufficiently small then, depending on ρ, one more step would probably give $\varepsilon_{k'+6} > 1-10^{-9}$.

To sum up, we have seen that the process is sure to increase ε_k above 1/10, and then for sufficiently small ρ that in six more steps ε_k is increased beyond $1-10^{-9}$.

The number of steps required to increase ε_k from 10^{-a} to 10^{-1} is about $\log_4 10^{a-1}$.

This completes our discussion of method II as modified for computation. Method I could easily be modified to be suitable for computation in a similar way. Without giving all details we may remark that the iterative formulas would be

(17.28)
$$F_o = 1$$
$$F_{k+1} = \alpha_k F_k (2 - AF_k)$$

or if the maximum proper value has been estimated (Cf. 17.)

$$F_{k+1} = \alpha_k F_k (2 - A_o F_k) .$$

Here again we must assume that $|A| < 1$.

Again the number α_k will have to be chosen as a real number slightly less than 1. Since in this case we will know nothing about the bound of A_o^{-1} the numbers ρ_k will have to be variable and be chosen at each step either by using the norm of F_k as an estimate for the bound, or by noticing that

$$|F_k| \leq 2^k .$$

The last remark is true because from (17.28) the bound of F_{k+1} is less than twice the bound of F_k.

18. Computation of $(M_1^* M_1)^{-1}$. The matrix A was obtained by the following steps:

$$B = M_1^* M_1 + X + n^2 \delta_o \cdot 1$$

$$\tau = t(B)$$

$$A = \frac{r}{\tau} B + S_1 \text{ , where } S_1 = rY + Z, \ |S_1| \leq 2n \delta_o$$

Then a matrix A_o was computed as follows:

$$A_o = \frac{A}{\lambda_o} + X_1 \text{ , } |X_1| \leq n \delta'' .$$

and we obtained an approximate value F_k for A_o^{-1}.

Therefore $\frac{1}{\lambda_o} F_k$ is an approximate value for A^{-1}, and $\frac{r}{\tau \lambda_o} F_k'$ approximate value for $(M_1^* M_1)^{-1}$.

Hence we have the following procedure:

First step: Compute

(18.1) $$\sigma = \frac{r}{\tau \lambda_o} + \eta_1 \qquad (\eta_1 \text{ is the round off error})$$

Second Step: Multiply F_k' by σ which leads to a matrix

(18.2) $$F'' = \sigma F_k' + Y'$$

We now have to estimate the difference between F'' and $(M_1^* M_1)^{-1}$, and we use the following inequality:

(18.3) $$\left| F'' - (M_1^* M_1)^{-1} \right| \leq \left| F'' - \frac{r}{\tau \lambda_o} F_k' \right| + \frac{r}{\tau \lambda_o} \left| F_k' - A_o^{-1} \right| + \left| \frac{r}{\tau \lambda_o} A_o^{-1} - (M_1^* M_1)^{-1} \right|$$

In order to estimate (18.1) notice that $\frac{1}{\lambda_o}$ has been computed before.

Moreover, the computation of $\frac{1}{\tau}$ introduces an additional error δ'' so that the computation of $\frac{r}{\tau\lambda_o}$ introduces an error of at most $(\frac{1}{\lambda_o} + 2)\delta''$, that is

$$|\eta_1| \leq (\frac{1}{\lambda_o} + 2)\delta'' . \tag{18.4}$$

If F'' is computed to the same number of significant digits as F_k', the error in one multiplication is smaller than

$$\left(\frac{r}{\tau\lambda_o} + 1\right) \delta''$$

and hence

$$|Y'| \leq n\delta * \left(\frac{r}{\tau\lambda_o} + 1\right) . \tag{18.5}$$

Assume, next, $|F_k' - A_o^{-1}| \leq \zeta$ so that

$$\frac{r}{\tau\lambda_o} |F_k' - A_o^{-1}| \leq \frac{r}{\tau\lambda_o} \zeta .$$

To estimate the last term in the inequality (18.3), observe that for any two matrices W_1, W_2, we have

$$W_1^{-1} - W_2^{-1} = W_1^{-1} (W_2 - W_1) W_2^{-1} \tag{18.6}$$

and hence

$$|W_1^{-1} - W_2^{-1}| \leq |W_1^{-1}| |W_2^{-1}| |W_1 - W_2| . \tag{18.7}$$

Observe also that

$$A_o = \frac{A}{\lambda_o} + X_1 = \frac{1}{\lambda_o} (\frac{r}{\tau} B + S_1) + X_1 = \frac{r}{\tau\lambda_o} B + \frac{S_1}{\lambda_o} + X_1 \tag{18.8}$$

$$= \frac{r}{\tau\lambda_o} (M_1 M_1 + X + n^2\delta_o) + \frac{S_1}{\lambda_o} + X_1 = \frac{r}{\tau\lambda_o} M_1 M_1 + L ,$$

where

$$L = \frac{r}{\tau\lambda_o} (X + n^2\delta_o \cdot 1) + \frac{S_1}{\lambda_o} + X_1$$

Hence:

$$|L| \leq 2n^2\delta_o \frac{r}{\tau\lambda_o} + \frac{2n\delta_o}{\lambda_o} + n\delta'' = \frac{2n^2\delta_o}{\tau\lambda_o} (1 + \frac{\tau}{n}) + n\delta''$$

From (18.8) it follows that $\frac{r}{\tau\lambda_o} \min (M_1 M_1) \geq A_o - |L|$.

Since for a positive definite matrix U, $|U^{-1}| = \frac{1}{\min U}$, we have

$$|(M_1^* M_1)^{-1}| \leq \frac{r}{\tau\lambda_o} \frac{1}{\min A_o - |L|} = \frac{r}{\tau\lambda_o} |A_o^{-1}| \frac{1}{1 - |A_o^{-1}||L|}$$

Hence

$$\left| \frac{r}{\tau\lambda_o} A_o^{-1} - (M_1^* M_1)^{-1} \right|$$

$$\leq \left| \left(\frac{\tau\lambda_o}{r} A_o \right)^{-1} \right| |(M_1^* M_1)^{-1}| \left| \frac{\tau\lambda_o}{r} A_o - M_1^* M_1 \right|$$

(18.9)

$$\leq \left(\frac{r}{\tau\lambda_o} |A_o^{-1}| \right)^2 \frac{1}{1 - |A_o^{-1}||L|} \frac{\tau\lambda_o}{r} |L|$$

$$= \frac{r}{\tau\lambda_o} |A_o^{-1}|^2 \frac{|L|}{1 - |A_o^{-1}||L|}$$

Combining (18.4), (18.6), (18.9), we find

$$|F'' - (M_1^* M_1)^{-1}|$$

$$\leq \frac{r}{\tau\lambda_o} \left\{ \left(1 + \frac{\tau\lambda_o}{r} \right) n\sigma'' + \xi + |A_o^{-1}|^2 \frac{|L|}{1 - |A_o^{-1}||L|} \right\} \tag{18.10}$$

For all practical purposes we may neglect the first term in the parenthesis (18.10) and we may replace $|L|$ by $\frac{2n^2\sigma_o}{\tau\lambda_o}$. Since $r < 1$, we finally have

$$|F'' - (M_1^* M_1)^{-1}| \leq \frac{\xi}{\tau\lambda_o} + \frac{|A_o^{-1}|^2}{(\tau\lambda_o)^2} \frac{2n^2\sigma_o}{1 - \frac{2|A_o^{-1}|n^2\sigma_o}{\tau\lambda_o}}$$

To the same degree of accuracy:

$$\frac{|A_o^{-1}|}{\lambda_o} = |A^{-1}|$$

hence

$$(18.11)\qquad \left|F'' - (M_1^* M_1)^{-1}\right| \le \frac{\zeta}{\tau \lambda_0} + \frac{\left|A^{-1}\right|^2}{\tau^2} \frac{2n^2\delta_0}{1 - \frac{2\left|A^{-1}\right| n^2 \delta_0}{\tau}}$$

We turn now to the computation of M_1^{-1}. Starting from the formula $(M_1^* M_1)^{-1} M_1^* = M_1^{-1}$, the last step is the multiplication of F'' by M_1^*, which leads to a matrix

$$(18.12)\qquad Q = F'' M_1^* + X_2$$

so that $Q - F'' M_1^* = X_2$, and $Q - M_1^{-1} = Q - F'' M_1^* + (F'' - (M_1^* M_1)^{-1})\ M_1^*$ that is

$$(18.13)\qquad \left|Q - M_1^{-1}\right| \le \left|X_2\right| + \left|F'' - (M_1^* M_1)^{-1}\right| \cdot \left|M_1^*\right|$$

where an estimate for $\left|F'' - (M_1^* M_1)^{-1}\right|$ is given by (18.11). If the multiplication in (18.12) is carried out to the same number of significant figures as the computation of F_k' we find

$$\left|X_2\right| \le \left(\frac{r}{\tau \lambda_0} + 1\right) n^2 \delta'', \text{ since } \left|M_1^*\right| \le 1,$$

so that, with sufficient accuracy,

$$(18.14)\qquad \left|Q - M_1^{-1}\right| \le \frac{\zeta}{\tau \lambda_0} + \frac{\left|A^{-1}\right|^2}{\tau^2} \frac{2n^2\delta_0}{1 - \frac{2\left|A^{-1}\right| n^2 \delta_0}{\tau}} + \frac{n^2 \delta''}{\tau \lambda_0}$$

In this formula we may insert any estimate for $\left|A^{-1}\right|$, that is it holds for both methods of inversion which have been discussed, but the greater the accuracy with which this is known, the better the estimate (18.14) becomes. Making use of the knowledge of the highest and lowest

Proper values we can replace $|A^{-1}|$ by $\frac{1}{\mu *}$ where $\mu *$ is a lower bound for the lowest proper value of A, which leads to

$$(18.15) \quad |Q - M_1^{-1}| \leq \frac{\xi}{\tau \lambda_0} + \frac{2n^2 \delta_0}{\mu *^2 \tau^2 (1-2\frac{n^2 \delta_0}{\mu * \tau})} \cdot \frac{n^2 \delta_n}{\tau \lambda_0}.$$

If $\mu *$ has not been computed, we have to rely on estimates based on the equation (16.1a). These, in general, will be less favorable and might lead to the sacrifice of a factor of order about n^2.

In (18.15) the third term on the right is negligible, but the first two terms are of comparable size as will now be shown. From remarks already made it is reasonable to expect $\mu *$ to be of order $1/n^3$ while τ might be assumed to be of order 1. The second term on the right of (18.15) therefore has order

$$(18.16) \quad n^8 \delta_0$$

This shows why it is necessary to choose δ_0 as small as possible.

We examine next the probable order of magnitude of the first term on the right of (18.15). From (17.25) we see that $1-A_0F_k'$ cannot be made smaller than 2ρ where

$$\rho = \frac{4n^5 \delta_n}{\varepsilon_0}$$

Thus we are forced to assume that ξ has order equal to

$$\frac{8n^5 \delta_n}{\varepsilon_0^2}$$

because $|A_0^{-1}|$ has order $(1/\varepsilon_0)$. The order of ε_0 is $1/n^2$ and the order of λ_0 is $1/n$ so the order of the first term on the right of (18.15) is

$$(18.17) \quad 8n^8 \delta_n .$$

From (18.16) and (18.17) we can estimate the probable number of decimal places which must be carried to get any desired accuracy,

remembering that for a more accurate estimate in any particular case we must return to (18.15). To evaluate these errors correctly it must be remembered that they are absolute errors and that $|M_1^{-1}|$ may be expected to be order $n^{3/2}$.

So far we have spoken only of round off errors and have assumed the matrix M to be given exactly. In any practical case the matrix elements are given only to a certain number of significant digits which means that they are given with a certain small error. We now add a few remarks about the accuracy with which the inverse is rigorously defined under these circumstances.

Let the theoretical value of the matrix be M_1 and let the matrix actually given be $M_1 + H$. If every element of H is at most η in absolute value then

$$|H| \leq n\,\eta \quad .$$

(If the matrix elements of H can be assumed to be statistically distributed then $|H|$ may be assumed of the order $n^{1/2}\,\eta$.)

Now

$$M_1 + H = M_1 (1 + M_1^{-1} H)$$

$$(M_1 + H)^{-1} = (1 + M_1^{-1} H)^{-1} M_1^{-1}$$

$$M_1^{-1} - (M_1 + H)^{-1} = (M_1^{-1} H)(1 + M_1^{-1} H)^{-1} M_1^{-1}$$

$$|M_1^{-1} - (M_1 + H)^{-1}| \leq \frac{|M_1^{-1}|^2 |H|}{1 - |M_1^{-1}||H|}$$

provided that $|M_1^{-1}||H| < 1$.

If it is required to have the difference (18.18) small compared to $|M_1^{-1}|$ then

$$\frac{|M_1^{-1}||H|}{1 - |M_1^{-1}||H|}$$

must be small compared to one, or what is equivalent

$$|M_1^{-1}| \quad |H|$$

must be small compared to one. It has been observed that M_1^{-1} may be expected to be of order $n^{3/2}$. Hence the order of $|M_1^{-1}|\,|H|$ will be $n^{5/2}\eta$ so that η must be small compared to $n^{-5/2}$. If the statistical assumption on H is made then η must be small compared to n^{-2}.

19. Summary. In this section we summarize the procedure to be used for the numerical computation of A, the extreme proper values of A, and the inverses of A and M. It may be convenient to use different accuracies in these various operations and our notation is as follows;

δ_o = round off error in computing A

δ = round off error in computation of the largest proper value

δ' = round off error in computation of smallest proper value

δ'' = round off error in computation of A^{-1}

We assume

$$\delta_o \leq \delta$$

a) Calculation of A

The elements of the matrix M are given to a certain number of significant figures. and δ_o is the maximum round off error in multiplying two numbers of absolute values less than one. If b is the base of the number system and p_o is the number of digits to the right of the decimal (binary) point then

$$\delta_o = (1/2)\, b^{-p_o}$$

Let β be the maximum absolute value of the elements m_{ij} and choose h such that

$$b^h \beta < (1/n)\ (1-n^2(n+1)\delta_o)^{1/2}$$

$$b^{h+1} \beta \geq (1/n)\ (1-n^2(n+1)\delta_o)^{1/2}$$

Then

(19.1) $$M_1 = b^h M$$

is formed without error. The next step is to calculate

(19.2) $$A = r\, \frac{M_1^* M_1 + n^2 \delta_o}{t\,(M_1^* M_1 + n^2 \delta_o)}$$

Here and elsewhere we do not attempt to indicate round off errors. As

has already been remarked it may be desirable to choose δ_0 considerably smaller than δ. The numbers r, n, and δ are required to satisfy

$$.9 \leq r \leq 1-4n^3\delta$$

(19.3) $$n^3\delta \leq 1/50$$

$$n \geq 3$$

b) Calculation of Maximum Proper value of A.

The following inductive procedure is to be followed:

(19.4) $$\begin{cases} B'_0 = A \\ B'_k = r \dfrac{B'^2_{k-1}}{t(B'^2_{k-1})} \end{cases}$$

Let

(19.5) $$n'_k = t(B'^2_{k-1})$$

and compute

(19.6) $$\sigma'_k = \frac{1}{2}\log_b n'_1 + \frac{1}{4}\log_b n'_2 + \cdots\cdots + \frac{1}{2^k}\log_b n'_k .$$

Then to guarantee that

(19.7) $$|\sigma'_k - \log_b r\lambda| < d$$

to a high degree of accuracy, we have only to require that

$$(1/2^k)\log_b (n/r) < d/2c$$

(19.8)

$$4\beta' + a\,[\,2k-3 + 3.2^{-k} + 2^k a\,(2.2 + 4(2^k a)\,] < \frac{d}{2}$$

where the symbols involved are defined as follows:

$$c = \log b \qquad \alpha^* = \frac{(1+\beta')}{r^2(1-\beta')^2}$$

(19.9) $$\beta' = 2n^2\delta/r$$

$$\alpha = 1.35\, n^{3/2}\delta \qquad a = n^2\alpha^* + n\beta'$$

When (19.8) and (19.7) are true then to a high degree of accuracy we will

have that the largest proper value λ and the computed largest proper value λ' satisfy

(19.10) $$(1-d)\lambda' < \lambda < (1+d)\lambda'$$

assuming that λ' is computed from σ_k' as indicated in (19.7).

c) Calculation of the minimum proper value of A.

Let

(19.11) $$\rho = \min [1-2n\delta, \lambda'(1+d)]$$

and compute

(19.12) $$H = \rho \cdot 1 - A$$

and

(19.13) $$G' = \frac{rH}{t(H)}$$

If λ_1 is the maximum proper value of G and μ the minimum proper value of A then

(19.14) $$\lambda_1 = \frac{\rho - \mu}{n\rho - r}$$

To estimate the error, let μ be the minimum proper value of A and let μ' be the computed value of μ. Let d' be the accuracy with which the maximum proper value of G has been computed and let

(19.15) $$d^* = (d' + 2n^2\delta')(\rho/\mu') + 4n\delta'/\mu' - d'$$

Then

(19.16) $$(1-d^*)\mu' \leq \mu \leq (1+d^*)\mu'$$

d) Calculation of A^{-1}.

Let μ and λ be the smallest and largest proper values of A and let μ^* and λ^* be numbers such that

$$0 < \mu^* \leq \mu, \quad \lambda \leq \lambda^* \leq 1.$$

Choose λ_0 such that

$$\lambda_0 \geq \frac{\lambda^*}{1 - n\delta''}$$

Calculate

(19.17) $$A_o = \frac{A}{\lambda_o}$$

so that

$$\min A_o \geq \frac{\mu^*}{\lambda_o} - n\delta^n = \varepsilon_o .$$

We assume $\varepsilon_o > 0$.

We carry out an inductive process described as follows:

$$F'_o = 1$$

(19.18) $$F'_{k+1} = \frac{4\alpha_k F'_k}{1+\varepsilon_k}\left(1 - \frac{A_o F_k}{1+\varepsilon_k}\right)$$

where ε_o has already been defined and ε_k, and α_k will now be defined in such a way that $0 < \varepsilon_k < 1$, $0 < \alpha_k < 1$. We wish to choose $\rho_k \geq |R_k|$ and for this we may choose ρ_k to satisfy

(19.19) $$\rho_k \geq n\delta^n \left\{4\left[n(n+1)\left|F'_k\right| + n\right] + 1\right\}$$

Then let

$$\alpha_k = 1 - \rho_k$$

(19.20) $$\varepsilon_k = \frac{4\alpha_{k-1}\varepsilon_{k-1}}{(1+\varepsilon_{k-1})^2} - \rho_{k-1}$$

In many cases it may be desirable to let ρ_k be constant and in fact

(19.21) $$\rho_k = \rho = 4n^3\delta^n/\varepsilon_o .$$

Then

$$\varepsilon_{k+1} \geq \frac{4\varepsilon_k}{(1+\varepsilon_k)^2} - 2\rho$$

(19.22) $$1 - \varepsilon_{k+1} \leq \left(\frac{1-\varepsilon_k}{1+\varepsilon_k}\right)^2 + 2\rho$$

which is a way of testing how near we are to a solution. We see also that since $1 - \varepsilon_k$ is the known estimate for $|1 - A_o F_k'|$ we can not expect to make this last quantity smaller than $2\rho = \dfrac{8n^3 \delta_n}{\varepsilon_o}$

Let

(19.23) $$\zeta \geq |F_k' - A_o^{-1}| .$$

We may choose

(19.24) $$\zeta = \frac{8n^3 \delta_n}{\varepsilon_o^2}$$

Then if $t = t\,(M_1^* M_1 + n^2 \delta_o I)$

(19.25) $$|Q - M_1^{-1}| \leq \frac{\zeta}{\tau \lambda_o} + \frac{2n^2 \delta_o}{\mu^{*2} \tau^2 \left(1 - 2\,\frac{n^2 \delta_o}{\mu^* \tau}\right)} + \frac{n^2 \delta_n}{\tau \lambda_o}$$

As a rough probable estimate we may say that the right hand side of (19.25) is at most

(19.26) $$n^8 \delta_o + 8n^8 \delta_n$$

and this will show approximately how many decimal places must be carried in general. To get exact information in any special case we must use (19.25).

In case we use the method of inversion in which we do not have information about the proper values, then the estimate (19.25) must be replaced by the formula (18.14) and in this case the result will not be as favorable as (19.26).

20. Examples. We list here some examples as an illustration of what may be expected to happen in inverting a matrix. We choose $n = 20$ and $n = 50$, and for each of these values of n, we choose two different values

of λ/μ , namely n^2 and $10n^2$. As has already been remarked it is reasonable to suppose that λ/μ is of the order n^2. We assume that μ is $1/n$ in every case so that the bound of M_1^{-1} is $n^{3/2}$ and $10^{1/2}n^{3/2}$ in the respective cases. The accuracy chosen for the inverse is 1 in 1000 compared to this bound. Without great expense this accuracy can be increased, for example if an accuracy of 1 in 10^5 is desired it is only necessary to take two more decimals and at most one more step in the process of finding the inverse. The results are displayed in the table on the following page.

Examples for Inverting Matrices

	M	20	20	50	50
	μ	1/20	1/20	1/50	1/50
	λ/μ	400	4000	2500	25000
	No. of decimals for A	12	14	15	17
LARGEST PROPER VALUE	Accuracy	1:10	1:10	1:10	1:10
	Steps	6	6	7	7
	Decimals	8	8	9	9
LOWEST PROPER VALUE	Accuracy	4:100	4:100	3:100	3:100
	Steps	16	19	20	23
	Decimals	11	12	13	14
	Accuracy in largest proper value of G required.	1:10000	1:100,000	1:100,000	1:1,000,000
INVERSE	Decimals	13	14	15	17
	Steps	8	10	10	12
	Relative accuracy	1:1000	1:1000	1:1000	1:1000
Total symmetric matrix multiplications		38	45	47	54
Total matrix multiplications		2	2	2	2

abv study