JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematical Sciences
Vol. 74B, No. 4, October–December 1970

Error Estimates for the Solution of Linear Algebraic Systems*

Brother Kenneth E. Fitzgerald, F.S.C.**

(July 20, 1970)

In this paper bounds for the error of a computed inverse of a matrix are developed. These are then applied to the solution of a single system. Methods for improving the approximate inverse are then discussed with some observations on the dangers involved in their practical use on a computer and some safeguards are indicated. Some computer programs for matrix inversion are then evaluated by means of the bounds developed.

Key words: Error estimates; evaluation of computer programs; inverse of a matrix; linear systems and matrices.

Contents

Introduction

The advent of modern high-speed electronic computers has led to the solution of many problems that had been considered almost unsolvable some years ago. The essential contribution of these computers is that they can do simple things extremely fast. Mathematical problems, like the solution of a system of many linear algebraic equations, can be done in a few seconds whereas it would take many hours by a person with a desk calculator and even longer if he used only pencil and paper. If we actually solved the problem in either of the latter ways, we would have a good idea of the errors we allowed to enter the process and could keep track of them. It is in this area that high-speed computers have some limitations. We feed numbers into a computer and the com-

AMS subject classification. Primary 65F35, Secondary 68A20.

*An invited paper. A dissertation submitted to the faculty of the Graduate School of Arts and Sciences of the Catholic University of America in partial fulfillment of the requirements for the degree Doctor of Philosophy.

**Present address: Manhattan College, Bronx, N.Y. 10471

puter prints out other numbers—supposedly the solution of our problem. But how do we know that these numbers *are* the solution of our problem? That this is a problem is pointed out very well by Leslie Fox:

> In the desk-machine era of Comrie and Peters, the great compilers of mathematical tables, the possibility of errors and the perpetual need for guarding against them were fully realised. They would be appalled by the self-confidence of their successors! [1, p. 16][1]

It certainly is possible that the machine could malfunction; some physical part could be defective—this has, in fact, happened and with no indication to the operator. Some modern text books on numerical analysis do not seem to recognize this possibility and this is most unfortunate [2, p. 111]. We will indicate in section 6 one way of checking for this kind of error.

Another possibility that exists is that our instructions to the machine (the "program") may not accomplish exactly what we had intended. There are many reasons for this, such as human error in punching the deck, the rounding process that continually occurs, subtraction of numbers that are very close to each other, and even poor programming. These difficulties are being investigated very seriously now by many men in the field, especially some in the Special Interest Group in Numerical Mathematics of the Association for Computing Machinery. We must be very sure that the computer program we are using does, in fact, perform successfully. Evaluation of existing computer programs is going on [3] and, in fact, we have devoted the second part of our research to this very thing, but, of course, only in a specific area. The criteria for this evaluation will be discussed at the beginning of section 6 and the search for these criteria gave the impetus for the first part of this research.

In this first part we derive some theoretical bounds for the accuracy of a computed solution of a system of linear algebraic equations. We concentrate first on the problem of finding an error bound for the computed inverse of a matrix (the solution of a set of linear algebraic systems) and then consider the solution of a single system. (This order may seem backwards at first, but we indicate its necessity at the end of section 2.) Our orientation will be to use what is computationally available. That is to say, if we are interested in the accuracy of a computed inverse, X, we will assume that the residual, $Y=I-AX$, is also available. We will then discuss ways of improving the computed inverse and the calculated solution of a single system and finally discuss error bounds for some special matrices.

In spite of the power of a high-speed computer, these theoretical relations discussed in the first part are of the utmost importance. We attempt to derive realistic (practical) bounds and we have opportunity in section 3 to show where the computed results must be very carefully examined; that is, they contradict our theoretical relations. Hence, our theoretical results can always be used at least in this negative way.

One of the concepts basic to a theoretical treatment of errors is that of a norm, and we feel it would be useful to give a brief discussion of this concept here, before proceeding to the body of the text. A full treatment of this topic can be found in *Computational Methods of Linear Algebra*, by Faddeev and Faddeeva. In dealing with vectors and matrices we frequently need some way of describing the concept of the size of a vector, x, or a matrix, A. We shall use the notation $N(x)$ or $N(A)$ to indicate this concept.

In two-dimensional Euclidean space, the size of a vector $x = (x_1, x_2)$ is just the length (always non-negative) of that vector:

$$N(x) = (x_1^2 + x_2^2)^{1/2}.$$

The length of such a vector has some immediate properties:

1. $N(x) > 0$ for $x \neq 0$ and $N(0) = 0$;
2. $N(cx) = |c|N(x)$ for any numerical multiplier c;
3. $N(x + y) \leq N(x) + N(y)$.

[1] Figures in brackets indicate the literature references on p. 309.

The usual length of a vector is just one way of describing its size. Other ways would depend upon our main point of emphasis. If we are interested in the distance in reference to the coordinate axes, we might take the norm of x to be the maximum distance from the origin along one of the axis:

$$N(x) = \max_i |x_i|.$$

For this definition, the same three properties also hold. The main thing, then, that we wish to associate with the concept of a norm is that it possess the three properties mentioned above. The generalization to n dimensions of each of the two examples given is obvious (although the proofs of some of the properties may not be), so let us consider the norm of a matrix.

DEFINITION: *The norm of a square matrix* A *is a nonnegative number* N(A) *which satisfies the conditions*

1. $N(A) > 0$, *if* $A \neq 0$ *and* $N(0) = 0$;
2. $N(cA) = |c|N(A)$;
3. $N(A + B) \leq N(A) + N(B)$
4. $N(AB) \leq N(A)N(B)$.

Some examples are in order. Similar to the length of a vector, we could take the square root of the sum of the squares of all the elements of A;

$$N(A) = \left(\sum_{i,j} |a_{ij}|^2\right)^{1/2} \qquad \text{(the Frobenius norm).}$$

Corresponding to the maximum element of a vector, we could take the maximum row sum of A:

$$N(A) = \max_i \sum_j |a_{ij}|.$$

Another one frequently used is also related to the maximum element of an $n \times n$ matrix:

$$N(A) = n \max_{i,j} |a_{ij}|.$$

In this case, the multiplier n is needed in order to satisfy the required properties.

Property 4 is peculiar to matrices and we have one special case to consider here; viz., the product Ax, a matrix times a vector. In this case we must choose the norms to be used in such a way that

$$N(Ax) \leq N(A)N(x).$$

This is called the compatibility condition. For each norm of a vector, there may be more than one norm of a matrix that can be used for $N(A)$ but not every matrix norm may satisfy this inequality.

Our main need for these norms arises when we consider the error of an approximate inverse X. We need some way to discuss the "size" of the error, $A^{-1} - X$, and the concept of a norm does this very well.

1. Error Bounds for the Approximate Inverse

In working with norms, frequent use if made of the property that $N(AB) \leq N(A)N(B)$. Before discussing various error bounds, it will be useful to examine this inequality, as it is the main reason why so many error estimates are overestimates. Let us take the Frobenius norm:

$$N(A) = \left(\sum_{i,j} |a_{ij}|^2\right)^{1/2}.$$

Then, for a general A and B,

$$\begin{aligned}
N(AB)^2 &= \sum_{i,j} \left| \sum_k a_{ik} b_{kj} \right|^2 \\
&\leqslant \sum_{i,j} \left(\sum_k |a_{ik} b_{kj}| \right)^2 \\
&= \sum_{i,j,r,s} |a_{ir} b_{rj} a_{is} b_{sj}| \\
&= \sum_{i,j,r,s} |a_{ir} b_{sj}| \, |a_{is} b_{rj}| \\
&\leqslant \sum_{i,j,r,s} \frac{1}{2} \left(|a_{ir} b_{sj}|^2 + |a_{is} b_{rj}|^2 \right) \\
&= \sum_{i,j,r,s} |a_{ir} b_{sj}|^2 \\
&= \left(\sum_{i,r} |a_{ir}|^2 \right) \left(\sum_{s,j} |b_{sj}|^2 \right) \\
&= N(A)^2 N(B)^2.
\end{aligned}$$

From this derivation we can determine when equality is possible. If we consider matrices with only non-negative elements, the first inequality in the above becomes an equality. Hence, the remaining inequality results from the fact that $2ab \leqslant a^2 + b^2$ for all a and b. This is an equality if and only if $a = b$. In our problem, this means if and only if $a_{ir} b_{sj} = a_{is} b_{rj}$ or if and only if

$$\frac{a_{ir}}{a_{is}} = \frac{b_{rj}}{b_{sj}}$$

which says that the columns of A are in the same ratio as the rows of B. This conclusion can be written as $b_{rj} = (b_{sj}/a_{is}) a_{ir}$ for all r; that is, each column of B is a multiple of a row of A. The last statement can be seen very clearly if we examine the Cauchy-Bunyakovskii inequality which is the basis for the step in the above derivation that we are discussing. This inequality states that for two vectors, X and Y, we have $|(X, Y)| \leqslant |X| \cdot |Y|$ where (X, Y) is the scalar product. The familiar proof is as follows. Let us consider a nonzero X, and let $Z = Y - aX$, where $a = (Y, X)/(X, X)$. We note that $(Z, X) = (Y, X) - a(X, X) = 0$. Then

$$\begin{aligned}
|Z|^2 = (Z, Z) &= (Z, Y - aX) = (Z, Y) = (Y - aX, \ Y) = (Y, Y) - a(X, Y) \\
&= (Y, Y) - \frac{(Y, X)(X, Y)}{(X, X)} = \frac{|X|^2 |Y|^2 - |(X, Y)|^2}{|X|^2}.
\end{aligned}$$

Consequently $|X|^2 |Y|^2 - |(X, Y)|^2 = |X|^2 |Z|^2 \geqslant 0$, and the desired inequality follows. We will have equality if and only if $Z = 0$. This says that $Y = aX$; i.e., one vector is a scalar multiple of the other.

Although this knowledge of when equality occurs does not give us any idea of the magnitude of the inequality, it certainly indicates that $N(AB)$ will seldom be equal to $N(A)N(B)$. Since we have no knowledge of when the elements of A and B are going to be non-negative, the first inequality in the derivation just makes matters worse. If the signs of the elements of A and B are such that cancellation takes place, this inequality could also cause a tremendous difference. It has been our experience in general that the norm of a product is very much less than the product of the norms. (See the example at the end of this section.)

There is one case, however, when equality will always hold if N is the Frobenius norm. If U and V are unitary matrices, then $N(UAV) = N(A)$. To see this, we use $N(A)^2 = \text{trace } (AA^*)$, A^*

being the conjugate transpose of A, and the fact that if two matrices are similar, their traces are equal (since the trace is equal to the sum of the eigenvalues). Thus

$$\begin{aligned} N(UAV)^2 &= tr(UAVV^*A^*U^*) \\ &= tr(UAA^*U^*) \\ &= tr(AA^*) \\ &= N(A)^2. \end{aligned}$$

This equality has the following two consequences. If A is a normal matrix, it is unitarily equivalent to a diagonal matrix, and we have $N(AB) = N(U^*DUB) = N(DUB) \leq N(D)N(UB) = N(D)N(B) \leq n^{1/2} \max |l_i| N(B)$ where l_i are the eigenvalues of A. The second one also concerns the eigenvalues of A. We use a result of Schur which states that an arbitrary complex matrix A can be transformed into triangular form by a unitary matrix U. Now for any triangular matrix T, we have

$$\sum_i |t_{ii}|^2 \leq N(T)^2$$

with equality if and only if T is diagonal. Hence,

$$\sum_i |l_i|^2 = \sum_i |t_{ii}|^2 \leq N(T)^2 = N(U^*AU)^2 = N(A)^2$$

with equality if and only if A is unitarily equivalent to a diagonal matrix; i.e., when A is normal.

As a final note on the relation between $N(AB)$ and $N(A)N(B)$, we would like to mention that there exists no norm such that equality holds for all A and B. To prove this, we would just consider an $A = B$ such that $A \neq 0$ but $A^2 = 0$.

Our main concern will be with bounds for $N(A^{-1})$ and $N(A^{-1} - X)$ where X is an approximate inverse and N is any norm. To derive upper bounds for $N(A^{-1})$ is a nontrivial matter. Since A may be arbitrarily close to a singular matrix, uniform upper bounds for $N(A^{-1})$ are not to be found. Hence, we must make use of information that can be readily available. The most useful quantity for an approximate inverse X is the residual matrix $Y = I - AX$. The "size" of Y, of course, depends on how good an approximation X is but we have found that $N(Y)$ is invariably less than 1 and, in fact, actually less than $\frac{1}{2}$.

The fact that $N(Y)$ is computationally less than 1 turns out to be quite important because this inequality is basic to practically all our theoretical results. Its main use follows from the following two lemmas.

LEMMA 1.1. *If* B *is a matrix such that* N(B) < 1, *then*

(a) (I ± B) *is nonsingular and*

(b) $N[(I \pm B)^{-1}] \leq \dfrac{1}{1-N(B)}$ *if* N(I) = 1.

PROOF: (a) Assume $(I \pm B)$ is a singular. Then there exists some vector $x \neq 0$ such that

$$(I \pm B)x = 0$$

or

$$x = \mp Bx$$

and

$$N(x) \leq N(B)N(x).$$

Since $N(x) > 0$, this last inequality means that $N(B) \geq 1$ which contradicts our condition that $N(B) < 1$.

(b) Since $(I \pm B)$ is nonsingular, we can let

$$R = (I \pm B)^{-1}.$$

Then

$$R^{-1} = (I \pm B)$$

$$I = R \pm RB$$

$$R = I \mp RB$$

and

$$N(R) \leq N(I) + N(R)N(B)$$

$$\leq 1 + N(R)N(B) \text{ if } N(I) = 1.$$

Then

$$N(R)[1 - N(B)] \leq 1$$

and

$$N(R) \leq \frac{1}{1 - N(B)}.$$

Another relation that needs the same condition and one that is also very useful is the following.

LEMMA 1.2. *If* B *is a matrix such that* N(B) < 1, *then*

$$(I - B)^{-1} = I + B + B^2 + \ldots + B^n + \ldots.$$

PROOF:

$$\text{Let } P_n = (I + B + B^2 + \ldots + B^{n-1})(I - B)$$

$$= I - B^n.$$

Then

$$I - P_n = B^n$$

and

$$N(I - P_n) = N(B^n)$$

$$\leq N(B)^n.$$

Since $N(B) < 1$, $N(B)^n$ approaches zero as n goes to infinity so that $N(I - P_n)$ approaches zero also and $I - P_n$ must converge to the 0 matrix, which means P_n converges to the identity matrix and hence $(I + B + B^2 + \ldots + B^{n-1})$ converges to $(I - B)^{-1}$.

With these two lemmas as background, we can proceed to derive the bounds for the matrices in which we are interested.

THEOREM 1.1. *Let* X *be an approximate inverse of* A *and let* Y = I − AX. *Then if* N(Y) < 1,

$$\frac{N(X)}{1 + N(Y)} \leq N(A^{-1}) \leq \frac{N(X)}{1 - N(Y)}.$$

PROOF: We shall prove the upper bound first.

$$Y = I - AX$$

$$AX = I - Y$$

$$X^{-1}A^{-1} = (I - Y)^{-1}$$

$$A^{-1} = X(I - Y)^{-1}$$

and

$$N(A^{-1}) \leq \frac{N(X)}{1 - N(Y)} \text{ if } N(I) = 1.$$

We may remove the restriction on $N(I)$ by the following:

$$A^{-1} = X(I - Y)^{-1}$$

$$= X(I + Y + Y^2 + Y^3 + \ldots)$$

$$= X + XY + XY^2 + XY^3 + \ldots$$

and

$$N(A^{-1}) \leq N(X) + N(X)N(Y) + N(X)N(Y^2) + \ldots$$
$$\leq N(X) + N(X)N(Y) + N(X)N(Y)^2 + \ldots$$
$$= N(X)[1 + N(Y) + N(Y)^2 + \ldots]$$
$$= \frac{N(X)}{1 - N(Y)}.$$

For the lower bound, we proceed in a similar way.

$$Y = I - AX$$
$$AX = I - Y$$
$$X = A^{-1}(I - Y)$$
$$= A^{-1} - A^{-1}Y$$

and

$$N(X) \leq N(A^{-1})(1 + N(Y))$$

so that

$$\frac{N(X)}{1 + N(Y)} \leq N(A^{-1}).$$

We note that taking $Y = I - XA$, the lefthand residual matrix, would yield identical results.

COROLLARY. *For any nonsingular* A, *let* D *be the diagonal and let* $-$M *represent the off-diagonal part of* A.

Then, if $N(D^{-1}M) < 1$,

$$\frac{N(D^{-1})}{1 + N(D^{-1}M)} \leq N(A^{-1}) \leq \frac{N(D^{-1})}{1 - N(D^{-1}M)}.$$

PROOF: If we write $A = D - M = D(I - D^{-1}M)$, then $A^{-1} = (I - D^{-1}M)^{-1}D^{-1}$ and the proof follows as in the theorem.

Let us now consider the error matrix $A^{-1} - X$, and derive some bounds for its norm.

THEOREM 1.2. *Let* X *be an approximate inverse of* A *and let* Y = I $-$ AX. *Then if* N(Y) < 1,

$$N(A^{-1} - X) \leq \frac{N(XY)}{1 - N(Y)}.$$

PROOF:

$$Y = I - AX$$
$$AX = I - Y$$
$$X^{-1}A^{-1} = (I - Y)^{-1}$$
$$= I + Y + Y^2 + Y^3 + \ldots$$
$$A^{-1} = X + XY(I + Y + Y^2 + Y^3 + \ldots)$$
$$A^{-1} - X = XY(I - Y)^{-1}$$

so that

$$N(A^{-1} - X) \leq \frac{N(XY)}{1 - N(Y)},$$

where we have omitted the restriction that $N(I) = 1$ since it can be eliminated by the same process as used in the proof of Theorem 1.1.

COROLLARY. *Under the same conditions as Theorem 1.2,*

$$N(A^{-1} - X) \leq \frac{N(X)N(Y)}{1 - N(Y)}.$$

PROOF: The proof follows immediately from the theorem and the properties of a norm.

We have developed this latter upper bound as a corollary since, as we mentioned at the beginning of this section, this bound is frequently very much larger than the one in the theorem. It is obviously simpler and demands very little additional computer time but the information may be misleading (see the example at the end of this section).

If we use the lefthand residual matrix, $Y_1 = I - XA$, the main difference from the above would be the need for $N(Y_1X)$ instead of $N(XY)$. However,

$$Y_1X = (I - XA)X = X - XAX = X(I - AX) = XY$$

so that actually the only difference comes from the use of $N(Y_1)$ instead of $N(Y)$. Unfortunately, it is not true in general that $Y_1 = Y$. In fact there are examples where there is a very large difference [4, p. 258]. So it is possible to have a good right inverse but a poor left inverse—good in the sense that the residual matrix is close to the zero matrix. Considering both residual matrices is of value, since they can give an indication of a poor inverse.

THEOREM 1.3. *Let* $E = (I - XA) - (I - AX)$. *Then*

$$N(A^{-1} - X) \geq \frac{N(E)}{2N(A)}.$$

PROOF:

$$E = (I - XA) - (I - AX)$$

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

$$N(E) \leq 2N(A)N(A^{-1} - X)$$

so that

$$N(A^{-1} - X) \geq \frac{N(E)}{2N(A)}.$$

So the difference between the residual matrices gives us a lower bound which provides us with negative, but still useful, information.

The question of which residual matrix to use in Theorem 1.2 and/or its corollary still remains. Using the theorem and requiring that the norm of both residuals be less than one, there should be very little difference. In general, however, the fact that the norm of one residual is small does not guarantee that the norm of the other will also be small:

$$I - XA = A^{-1}(I - AX)A$$

so that

$$N(I - XA) \leq N(A)N(A^{-1})N(I - AX)$$

and similarly

$$N(I - AX) \leq N(A)N(A^{-1})N(I - XA).$$

For an ill-conditioned system, $N(A)N(A^{-1})$, the condition number, can be quite high. In practice, however, we find that the two residuals are almost invariably quite close, so that using Theorem 1.2 with either residual should yield satisfactory results. Since inverses are calculated mostly in such a way as to almost guarantee the smallness of $I - AX$, it would seem that using $I - XA$ in our error bounds would give more realistic information since we really expect a two-sided inverse. We also note that if it were particularly important to have $I - XA$ small, it would be safer to invert A^T, the transpose of A.

We have been considering only problems where the coefficient matrix had exact numerical entries. We would like to give now an error bound for the calculated inverse where there is some uncertainty in A [5]. Although we are working with the matrix A, we should really be working with A^*, which differs from A by an amount satisfying $N(A-A^*) \leq a$. We must, therefore, include this known difference in our bound for $N((A^*)^{-1}-X)$.

THEOREM 1.4. *Let* $\mathrm{N(A - A^*)} \leq \mathrm{a}$ *and let* $\mathrm{R = I - A^*X}$. *Then if* $\mathrm{N(R)} < 1$,

$$\mathrm{N((A^*)^{-1} - X)} \leq \frac{\mathrm{N(X)[N(I - AX) + aN(X)]}}{1 - \mathrm{N(I - AX)} - \mathrm{aN(X)}}.$$

PROOF: We proceed exactly as in Theorem 1.2 and use R instead of Y, arriving at

$$N((A^*)^{-1} - X) \leq \frac{N(X)N(R)}{1 - N(R)}.$$

Now $R = I - A^*X = I - AX + AX - A^*X = (I - AX) + (A - A^*)X$ and $N(R) \leq N(I - AX) + N(A - A^*)N(X) \leq N(I - AX) + aN(X)$. Substituting this expression in the above inequality, we have the result stated in the theorem.

Let us now turn our attention to the relative error of the difference between A^{-1} and X.

THEOREM 1.5. *Let* X *be an approximate inverse of* A *and let* $\mathrm{Y = I - AX}$. *If* $\mathrm{N(Y)} < 1$,

$$\frac{1 - \mathrm{N(Y)}}{\mathrm{N(X)}} \leq \frac{1}{\mathrm{N(A^{-1})}} \leq \frac{1 + \mathrm{N(Y)}}{\mathrm{N(X)}}.$$

PROOF: The proof follows immediately from Theorem 1.1.

COROLLARY. $$1 - \mathrm{N(Y)} \leq \frac{\mathrm{N(X)}}{\mathrm{N(A^{-1})}} \leq 1 + \mathrm{N(Y)}.$$

THEOREM 1.6. *Let* X *be an approximate inverse of* A *and let* $\mathrm{Y = I - AX}$. *If* $\mathrm{N(Y)} < 1$,

$$\frac{\mathrm{N(A^{-1} - X)}}{\mathrm{N(A^{-1})}} \leq \frac{\mathrm{N(XY)}}{\mathrm{N(X)}} \cdot \frac{1 + \mathrm{N(Y)}}{1 - \mathrm{N(Y)}}.$$

PROOF: The proof follows directly from Theorem 1.2 and Theorem 1.5.

COROLLARY. $$\frac{\mathrm{N(A^{-1} - X)}}{\mathrm{N(A^{-1})}} \leq \mathrm{N(Y)} \cdot \frac{1 + \mathrm{N(Y)}}{1 - \mathrm{N(Y)}}$$

Actually, we have a simpler and better bound than this corollary.

THEOREM 1.7. *Let* X *be an approximate inverse of* A *and let* $\mathrm{Y = I - AX}$ *or* $\mathrm{I - XA}$. *Then*

$$\frac{\mathrm{N(A^{-1} - X)}}{\mathrm{N(A^{-1})}} \leq \mathrm{N(Y)}.$$

PROOF: $A^{-1} - X = A^{-1}(I - AX) = (I - XA)A^{-1}$ so that after taking norms, the inequaltiy follows.

This bound is not necessarily smaller than the one given in Theorem 1.6. If $N(Y)$ is small enough (say less than .1), the ratio in the above corollary can be expanded to the product

$$[1 + N(Y)][1 + N(Y) + N(Y)^2 + \ldots]$$

or to $1 + 2N(Y)$ if we ignore the higher powers of $N(Y)$. Then the corollary becomes

$$\frac{N(A^{-1} - X)}{N(A^{-1})} \leq N(Y)[1 + 2N(Y)] \approx N(Y),$$

again ignoring the higher powers of $N(Y)$. This indicates that in practice the bounds given in Theorem 1.7 and the corollary are quite close so that the bound given in Theorem 1.6 will probably still be the smaller one.

Finally, we also have a lower bound.

THEOREM 1.8. *Let* X *be an approximate inverse,* $Y = I - AX$, *and let* $E = (I - XA) - (I - AX)$. *Then if* $N(Y) < 1$,

$$\frac{N(A^{-1} - X)}{N(A^{-1})} \geqslant \frac{N(E)}{2N(A)} \cdot \frac{1 - N(Y)}{N(X)}.$$

PROOF: The proof follows immediately from Theorem 1.3 and Theorem 1.5.

We conclude this section with a numerical example of the above bounds. Let N be the maximum row sum norm and let A be the 6×6 segment of the infinite Hilbert matrix A, where $a_{ij} = 1/(i+j-1)$. We chose this particular matrix because of its notoriously poor condition number, about 2.92×10^8, but using only a 6×6 so that roundoff errors would not be too influential. Since we will use this matrix in all our examples, it might be good to give its *exact* inverse. For the nth order segment, if $A_n^{-1} = (b_{ij}^{(n)})$, then

$$b_{ij} = \frac{(-1)^{i+j}(n+i-1)!(n+j-1)!}{(i+j-1)[(i-1)!(j-1)!]^2(n-i)!(n-j)!}.$$

In our case (giving only the lower triangular part since it is symmetric) this turns out to be

$$A_6^{-1} = \begin{bmatrix} 36 & & & & & \\ -630 & 14700 & & & & \\ 3360 & -88200 & 564480 & & & \\ -7560 & 211680 & -1411200 & 3628800 & & \\ 7560 & -220500 & 1512000 & -3969000 & 4410000 & \\ -2772 & 83160 & -582120 & 1552320 & -1746360 & 698544 \end{bmatrix}$$

We note that this is not quite the inverse of the matrix actually used in the examples. It is impossible to represent all elements $1/(i+j-1)$ exactly in the computer and hence the input matrix is slightly different. For further information on this point of inexact data of the Hilbert matrix, see reference 12 (*Computer Solution of Linear Algebraic Systems* by Forsythe and Moler), section 19. We also indicate in section 6 how to avoid this problem.

To calculate X we ran the program given in the above text (p. 68). The computer we used was a Univac 1108 and we used double precision for all matrix products.

A word about notation. The standard computer notation $.235\,E+04$ means $.235 \times 10^4$ or 2350. The "E" means exponent to the base 10 and the "+04" is the positive exponent.

The absolute error: $(Y = I - AX)$

$$(1)\quad N(A^{-1} - X) \leq \frac{N(X)N(Y)}{1 - N(Y)} = .197\,E+06$$

$$(2)\quad N(A^{-1} - X) \leq \frac{N(XY)}{1 - N(Y)} = .612\,E-01$$

$$(3)\quad N(A^{-1}-X) \geqslant \frac{N(E)}{2N(A)} \qquad = .477\ E-02$$

The relative error:

$$(1')\quad \frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant N(Y) \cdot \frac{1+N(Y)}{1-N(Y)} = .168\ E-01$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant N(Y) \qquad = .162\ E-01$$

$$(2')\quad \frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant \frac{N(XY)}{N(X)} \cdot \frac{1+N(Y)}{1-N(Y)} = .522\ E-08$$

$$(3')\quad \frac{N(A^{-1}-X)}{N(A^{-1})} \geqslant \frac{N(E)}{2N(A)} \cdot \frac{1-N(Y)}{N(X)} = .393\ E-09$$

and for

$$\frac{N(X)}{1+N(Y)} \leqslant N(A^{-1}) \leqslant \frac{N(X)}{1-N(Y)},$$

$$.117E+08 \leqslant N(A^{-1}) \leqslant .121E+08$$

where $N(A^{-1}) = .11931731\ E+08$.

Note: If we had been able to input the data exactly, the norm of the inverse would have been .11865420 E+08.

It might be of interest to give the values of the previous quantities using the left hand residual, $I-XA$, for Y.

The absolute error:

$$(1)\quad N(A^{-1}-X) \leqslant \frac{N(X)N(Y)}{1-N(Y)} \qquad = .380\ E+06$$

$$(2)\quad N(A^{-1}-X) \leqslant \frac{N(YX)}{1-N(Y)} \qquad = .622\ E-01$$

$$(3)\quad N(A^{-1}-X) \geqslant \frac{N(E)}{2N(A)} \qquad = .477\ E-02$$

The relative error:

$$(1')\quad \frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant N(Y)\ \frac{1+N(Y)}{1-N(Y)} \qquad = .328\ E-01$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant N(Y) \qquad = .309\ E-01$$

$$(2')\quad \frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant \frac{N(YX)}{N(X)}\ \frac{1+N(Y)}{1-N(Y)} = .537\ E-08$$

$$(3')\quad \frac{N(A^{-1}-X)}{N(A^{-1})} \geqslant \frac{N(E)}{2N(A)}\ \frac{1-N(Y)}{N(X)} = .387\ E-09$$

and for

$$\frac{N(X)}{1+N(Y)} \leq N(A^{-1}) \leq \frac{N(X)}{1-N(Y)},$$

$$.116\,E+08 \leq N(A^{-1}) \leq .123\,E+08.$$

We also note:

$$N(XY) = .6024\,E-01$$

$$N(YX) = .6024\,E-01 \qquad \text{and} \qquad N(E) = .2337\,E-01.$$

It will also be of interest to give the values of the previous quantities using a different computer, the CDC 6400, with $Y=I-AX$. This machine uses about twice as many digits in its operations as the Univac 1108 and this is why the results are so much better. We found that the values of the elements of X were integers (to 8 digits).

The absolute error:

$$(1)\ N(A^{-1}-X) \leq \frac{N(X)N(Y)}{1-N(Y)} = .323\,E-01$$

$$(2)\ N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} = .178\,E-07$$

$$(3)\ N(A^{-1}-X) \geqslant \frac{N(E)}{2N(A)} = .555\,E-09$$

The relative error:

$$(1')\ \frac{N(A^{-1}-X)}{N(A^{-1})} \leq N(Y)\,\frac{1+N(Y)}{1-N(Y)} = .272\,E-08$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq N(Y) = .272\,E-08$$

$$(2')\ \frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{N(X)}\,\frac{1+N(Y)}{1-N(Y)} = .150\,E-14$$

$$(3')\ \frac{N(A^{-1}-X)}{N(A^{-1})} \geqslant \frac{N(E)}{2N(A)}\,\frac{1-N(Y)}{N(X)} = .468\,E-16$$

and for

$$\frac{N(X)}{1+N(Y)} \leq N(A^{-1}) \leq \frac{N(X)}{1-N(Y)},$$

$$.1186542\,E+08 \leq N(A^{-1}) \leq .1186542\,E+08.$$

We also note that $N(XY) = .178\,E-07$ and $N(E) = .272\,E-08$.

2. Error Bounds for the Approximate Solution Vector of $Ax=b$

In this chapter we shall derive estimates for the closeness of the calculated solution vector to the true solution vector of $Ax=b$. Our approach will be to assume that we have available both the approximate inverse X and its residual $Y=I-AX$ or $I-XA$. The necessity for this will be explained by the quotations at the end of this section.

Let z be the approximate solution vector of $Ax=b$ and let $r=b-Az$, the residual vector.

THEOREM 2.1. *The bound for the norm of the error is given by*

$$\mathrm{N(A^{-1}b-z)} \leq \frac{\mathrm{N(X)N(r)}}{1-\mathrm{N(Y)}}, \qquad \mathrm{N(Y)} < 1.$$

PROOF:

$$b-Az=r$$

$$A^{-1}b-z=A^{-1}r$$

so that

$$N(A^{-1}b-z) \leq N(A^{-1})N(r)$$

$$\leq \frac{N(X)N(r)}{1-N(Y)}$$

using our estimates for $N(A^{-1})$ from the previous section.

THEOREM 2.2. *Let* $\mathrm{z=Xb}$. *Then if* $\mathrm{N(Y)} < 1$,

$$N(A^{-1}b-z) \leq \frac{N(XY)N(b)}{1-N(Y)}.$$

PROOF:

$$A^{-1}b-z=A^{-1}b-Xb$$

$$=(A^{-1}-X)b$$

and

$$N(A^{-1}b-z) \leq \frac{N(XY)N(b)}{1-N(Y)}$$

again using information from the previous section.

COROLLARY. $N(A^{-1}b-z) \leq \dfrac{N(X)N(Y)N(b)}{1-N(Y)}$.

We would like to make a remark stemming from the proof of Theorem 2.1. If the equality $A^{-1}r=A^{-1}b-z$ is ever used in practice, say for checking a process or computer program where the solution is known, we should note that we would be using X and not A^{-1}. That is to say

$$A^{-1}r=A^{-1}(b-Az)=A^{-1}b-A^{-1}Az=A^{-1}b-Iz=A^{-1}b-z$$

but

$$Xr=X(b-Az)=Xb-XAz=Xb-(I-Y_1)z=Xb-z+Y_1z,$$

where Y_1 is the lefthand residual $I-XA$. Hence, we have an additional factor to keep in mind, Y_1z. The need for this obviously depends on the quality of X; i.e., the closeness of XA to I. There are examples, however, where X may be a good inverse but where Xb may not be a good approximation to the solution vector [1, p. 149]. So, in general, the correction factor Y_1z can be of impor-

406-680 O - 71 - 3

tance. To give some indication of just how important, let us give an example.

In solving $Ax=b$ where A was the 6×6 Hilbert segment and b was $(1,2,3,4,5,6)^T$, we used Gaussian Elimination (with iterative improvement) to get X and the machine was an IBM 7094. For the maximum element norm, $N(Xr)$ was of order 10^{-2}; $N(Y_1z)$, of order 10^{-1}; $N(Xb-z)$, of order 10^{-1}. The need for the correction factor is clear. The fact that the Hilbert matrix is ill-conditioned results in the high order of the norms but the need is still there to get equality even for a well-conditioned matrix. Using a diagonally dominant tridiagonal matrix for A, we found that $N(X_r)$ was of order 10^{-7}; $N(Y_1z)$, of order 10^{-8}; $N(Xb-z)$, of order 10^{-8}. The need for the correction factor is again clear although not so vital. We would also like to emphasize that the residual to be used is the left hand one. As shown in section 1, there can be an appreciable difference in the two.

The use of the lefthand residual (only) also gives us a bound for the relative error.

THEOREM 2.3. *Let* $z=Xb$. *Then*

$$\frac{N(A^{-1}b-z)}{N(A^{-1}b)} \leq N(I-XA).$$

PROOF:

$$A^{-1}b-z=A^{-1}b-Xb$$

$$=(A^{-1}-X)b$$

$$=(I-XA)A^{-1}b$$

so that

$$\frac{N(A^{-1}b-z)}{N(A^{-1}b)} \leq N(I-XA).$$

Once again this residual seems to be of more practical use than $I-AX$. (It was noted in the previous section that $I-AX$ is ordinarily small by reason of the numerical procedure so that checking $N(I-XA)$ is the important thing.) We can see that a small right residual $I-AX$ may not guarantee the smallness of the error vector from the following:

$$A^{-1}b-z=A^{-1}b-Xb=(A^{-1}-X)b=A^{-1}(I-AX)b.$$

Hence, if A^{-1} has a large norm, the norm of the error vector may not be small.

It can also be seen from the proof of Theorem 2.1 that a small residual vector may *not* indicate a good solution vector:

$$A^{-1}b-z=A^{-1}r$$

so that

$$N(A^{-1}b-z) \leq N(A^{-1})N(r).$$

Again, if $N(A^{-1})$ is large, $N(A^{-1}b-z)$ could be large also. On the other hand, a large residual vector can indicate a poor solution vector. We show this in the following theorem.

THEOREM 2.4. *If* $r=b-Az$, *then*

$$N(A^{-1}b-z) \geq \frac{N(r)}{N(A)}.$$

PROOF:

$$r=b-Az$$

$$=A(A^{-1}b-z)$$

$$\frac{N(r)}{N(A)} \leq N(A^{-1}b-z).$$

We would like to include here a brief discussion of an error bound for the computed solution vector of a system that does not have exact data [5, p. 151]. As in Theorem 1.4, let

$$N(A-A^*) \leq a,\ N(b-b^*) \leq c,\ x^* = A^{-1}b,\ x^{**} = (A^*)^{-1}b^* \text{ and } z = Xb,$$

where X is the approximate inverse of A.

THEOREM 2.5. *Let* $\mathrm{R} = \mathrm{I} - \mathrm{A}^*\mathrm{X}$. *Then if* $\mathrm{N(R)} < 1$,

$$\mathrm{N}(z - x^{**}) \leq \frac{\mathrm{N(X)}[a\mathrm{N}(z) + \mathrm{N}(b - \mathrm{Az}) + c]}{1 - \mathrm{N(I - AX)} - a\mathrm{N(X)}}.$$

PROOF:
$$z - x^{**} = z - (A^*)^{-1}b^*$$
$$= (A^*)^{-1}(A^*z - b^*)$$

and
$$N(z - x^{**}) \leq N[(A^*)^{-1}]N(A^*z - b^*).$$

Now
$$A^*z - b^* = A^*z - Az + Az - b + b - b^*$$

so that
$$N(A^*z - b^*) \leq N(A^* - A)N(z) + N(Az - b) + N(b - b^*).$$

To get a bound for $N[(A^*)^{-1}]$ we proceed as in section 1, starting with $R = I - A^*X$, which leads to

$$N[(A^*)^{-1}] \leq \frac{N(X)}{1 - N(R)}.$$

Also
$$R = I - A^*X$$
$$= I - AX + AX - A^*X$$

and
$$N(R) \leq N(I - AX) + N(A - A^*)N(X).$$

Combining all of the above, we have the result stated in the theorem.

As mentioned at the beginning of the section, we have used the approximate inverse in all our bounds. This is most unfortunate because it means that we have to solve an additional n systems to get information about the solution of one system. It seems, however, that to expect any improvement in this situation is rather futile, according to some of the best men in the field. We shall quote from a few of them:

A. M. Turing: (1948) "It is difficult to determine the accuracy of the solution of a set of equations without inverting the matrix." [6, p. 301]

J. H. Wilkinson: (1963) "It does not seem possible to obtain a rigorous estimate of the accuracy of a solution to a set of equations without some estimate for $N(A^{-1})$. . . . To achieve more . . . we need an approximate inverse and a bound for its error." [7, p. 126]

E. L. Albasiny: (1964) "Unless one has some knowledge of the inverse of A we cannot immediately tell however how accurate our solution is." [8, vol. 1, p. 171]

L. Fox: (1965) "The search for the worst combination of uncertainties involves the computation of the inverse, and in solving linear equations we need a simpler and reasonably satisfactory criterion for the number of quotable figures in the calculated solution. If the coefficients vary widely in magnitude there is unlikely to be any satisfactory analysis . . ." [1, p. 159]

A. S. Householder: (1965) "For error estimates it is advantageous to find at least an approximation to A^{-1} even when only a single vector x is required." [9, p. 91]

W. Kahan: (1966) "The only disadvantage that can be occasioned by the lack of an estimate Z of A^{-1} is that there is no other way to get a rigorous errorbound for $z-x$." [10, p. 785]

C. Moler: (1967) ". . . without a bound for the condition of A, no precise conclusion can be made regarding $N(x_n-x)$." [11, p. 320]

Hence, it would seem that the bounds given in this section are about the only ones possible.

In some special cases, however, where A^{-1} is easily obtained or estimated, some improvement in this situation might be hoped for. We will discuss some of these in section 5.

3. Improvement Schemes for the Approximate Inverse

In this section we will present three schemes for improving the inverse that we have found quite effective. Each has a different order of convergence but not without some expense. We then generalize to a scheme of any order of convergence. Next, we show how two iteration methods which are frequently used are special cases of our schemes. Finally we give a numerical example of the various bounds derived.

The first scheme follows directly from the same procedure that yielded our error bounds for the approximate inverse in section 1. Let X be the approximate inverse and let $Y=I-AX$. Then

$$A^{-1}=X+XY+XY^2+\ldots.$$

The righthand side of this equality gives us our first scheme:

$$X_k=X+XY+XY^2+\ldots+XY^k, \qquad k\geq 0.$$

If $N(Y)<1$, we have convergence as k goes to infinity and by taking enough terms we can get as close to the true inverse as we please. The scheme is ideal for computer execution, and double precision (at least) should be used for the products of the matrices and also for the sum. The result would actually be a double precision inverse and we have found this inverse to be very effective in yielding $A^{-1}b$, the actual solution vector for a linear algebraic system.

The same scheme can give us error bounds for the closeness of X to A^{-1}. We have

$$A^{-1}-X=XY+XY^2+\ldots$$

so that we can get an excellent estimate for the difference just by taking enough terms. Hence, given any X we can tell how good it is, the only requirement being that $N(Y)<1$. If that particular X is not good enough, we can take more terms from the series until we are satisfied:

$$\begin{aligned} A^{-1}-X_k&=XY^{k+1}+XY^{k+2}+\ldots \\ &=XY^{k+1}(I+Y+Y^2+\ldots) \end{aligned}$$

so that

$$\begin{aligned} N(A^{-1}-X_k)&\leq\frac{N(XY^{k+1})}{1-N(Y)} \\ &\leq\frac{N(X)N(Y^{k+1})}{1-N(Y)} \\ &\leq\frac{N(X)N(Y)^{k+1}}{1-N(Y)}. \end{aligned}$$

As mentioned previously, the first bound is by far the best. However, the last bound can be quite useful. Having calculated X and Y, we can easily determine k to satisfy our requirements on

$N(A^{-1}-X)$ (realizing that this is definitely an overestimate), and proceed to sum the correct number of terms from our series and also determine the kind of multiprecision that may be needed to do the calculation correctly. Using this last bound also eliminates one matrix multiplication, to get XY^{k+1}.

If we use the lefthand residual, the procedure is essentially the same and we have for

$$N(I-XA)=N(Y)<1,$$

$$A^{-1}=X+YX+Y^2X+\ldots$$

$$X_k=X+YX+Y^2X+\ldots+Y^kX$$

and

$$N(A^{-1}-X_k)\leqslant\frac{N(Y^{k+1}X)}{1-N(Y)}$$

$$\leqslant\frac{N(X)N(Y^{k+1})}{1-N(Y)}$$

$$\leqslant\frac{N(X)N(Y)^{k+1}}{1-N(Y)}.$$

The second scheme we are interested in is very similar but the residual matrix must be recalculated at each step. We start as before but now let $X_0=X$ and define a recursion formula:

$$X_k=X_{k-1}+XY^k=X+XY+XY^2+\ldots+XY^k$$
$$Y_k=I-AX_k,\qquad k>0.$$

If $N(Y)<1$, we are assured of convergence as k goes to infinity and with Y_k as defined we proceed exactly as in section 1 to get

$$N(A^{-1}-X_k)\leqslant\frac{N(X_kY_k)}{1-N(Y_k)},\qquad N(Y_k)<1.$$

In our derivation for this bound we assumed X_k^{-1} exists. It is clear that one of our basic assumptions from the beginning has been that X^{-1} exists. It seems, however, that we should be more explicit about the existence of X_k^{-1}.

$$X_k=X+XY+XY^2+\ldots+XY^k$$
$$=X(I+Y+Y^2+\ldots+Y^k).$$

Now

$$(I-Y)^{-1}=I+Y+Y^2+\ldots+Y^k+Y^{k+1}+\ldots.$$

Hence

$$I+Y+Y^2+\ldots+Y^k=(I-Y)^{-1}-Y^{k+1}(I+Y+Y^2+\ldots)$$

$$=(I-Y)^{-1}-Y^{k+1}(I-Y)^{-1}$$

$$=(I-Y^{k+1})(I-Y)^{-1}.$$

Since $N(Y^{k+1})\leqslant N(Y)^{k+1}\leqslant N(Y)<1$, $(I-Y^{k+1})^{-1}$ exists and hence we are assured of the existence of X_k^{-1}.

In this scheme the residual actually gets multiplied by itself as the process continues:

$$\begin{aligned} Y_1 &= I - AX_1 \\ &= I - A(X + XY) \\ &= Y - AXY \\ &= (I - AX)Y \\ &= Y^2. \end{aligned}$$

and

$$\begin{aligned} Y_k &= I - AX_k \\ &= I - A(X_{k-1} + XY^k) \\ &= Y_{k-1} - AXY^k \\ &= Y^k - AXY^k \quad \text{(by induction)} \\ &= Y^{k+1}. \end{aligned}$$

Since $N(Y_k) \leq N(Y)^{k+1} < N(Y)$, our condition for convergence and the validity of the error bound is still $N(Y) < 1$.

Now $$X_k = X + XY + XY^2 + \ldots + XY^k$$

so that $$N(X_k) \leq N(X)[1 + N(Y) + N(Y)^2 + \ldots + N(Y)^k].$$

Since $$\frac{1 - p^{k+1}}{1 - p} = 1 + p + p^2 + \ldots + p^k, \qquad p \neq 1,$$

we have $$N(X_k) \leq N(X) \cdot \frac{1 - N(Y)^{k+1}}{1 - N(Y)}.$$

We also have $$\begin{aligned} X_k Y_k &= (X + XY + XY^2 + \ldots + XY^k)Y^{k+1} \\ &= XY^{k+1} + XY^{k+2} + \ldots + XY^{2k+1} \end{aligned}$$

so that $$\begin{aligned} N(X_k Y_k) &\leq N(XY^{k+1})[1 + N(Y) + \ldots + N(Y)^k] \\ &= N(XY^{k+1}) \frac{1 - N(Y)^{k+1}}{1 - N(Y)}. \end{aligned}$$

Hence $$\begin{aligned} \frac{N(X_k Y_k)}{1 - N(Y_k)} &\leq \frac{N(X_k Y_k)}{1 - N(Y)^{k+1}} \\ &\leq \frac{N(XY^{k+1})}{1 - N(Y)} \end{aligned}$$

and the best bound for our second scheme is better than that for our first.

Summarizing the bounds for the second scheme, we have for $N(Y) < 1$,

$$N(A^{-1}-X_k) \leqslant \frac{N(X_kY_k)}{1-N(Y_k)}$$

$$\leqslant \frac{N(X_k)N(Y)^{k+1}}{1-N(Y)^{k+1}}$$

$$\leqslant \frac{N(X)N(Y)^{k+1}}{1-N(Y)}.$$

It will be noticed that the last bound is exactly the same as for the first scheme, so that any improvement must come by using the first bound.

In spite of the fact that we recalculate the residual matrix at each step, we used it only in the error bound expression. It would seem useful to try to get it involved in the calculation of the new approximate inverse also. We can do this quite simply and very effectively by returning to the basic relation at each step. The process would be as follows:

$$A^{-1}=X+XY+XY^2+\ldots$$

with convergence if $N(Y) < 1$.

$$X_1=X+XY=X(I+Y)$$

$$Y_1=I-AX_1=Y^2.$$

Using this expression for Y_1, we repeat:

$$A^{-1}=X_1+X_1Y_1+X_1Y_1^2+\ldots$$

with convergence if $N(Y_1) < 1$

but $N(Y_1) \leqslant N(Y)^2$.

$$X_2=X_1+X_1Y_1=X_1(I+Y_1)$$

$$Y_2=I-AX_2=Y_1^2=Y^4=Y^{2^2}.$$

In general we have

$$X_{r+1}=X_r(I+Y_r)$$

$$Y_{r+1}=I-AX_{r+1}.$$

Now

$$\begin{aligned} Y_{r+1} &= I-A(X_r(I+Y_r)) \\ &= I-AX_r-AX_rY_r \\ &= Y_r-AX_rY_r \\ &= Y_r^2 \end{aligned}$$

so that

$$Y_r=Y^{2^r}.$$

Hence, if $N(Y) < 1$ we have convergence for our third scheme and since the error matrix is squared at each step, we have quadratic convergence.

It is interesting to observe that this third scheme actually amounts to taking a very large number of terms in our first series at each step. For example,

$$\begin{aligned} X_3 &= X_2(I+Y_2) \\ &= X_1(I+Y_1)(I+Y_1^2) \\ &= X(I+Y)(I+Y^2)(I+Y^{2^2}) \\ &= X+XY+XY^2+\ldots+XY^7 \end{aligned}$$

and in general,

$$\begin{aligned} X_r &= X(I+Y)(I+Y^2)\ \ldots\ (I+Y^{2^{r-1}}) \\ &= X+XY+XY^2+\ldots+XY^{2^r-1}. \end{aligned}$$

The recurrence formula, however, is much simpler and more efficient.

The bounds for the third scheme follow directly as before. If $N(Y)<1$,

$$\begin{aligned} N(A^{-1}-X_r) &\leqslant \frac{N(X_rY_r)}{1-N(Y_r)} \\ &\leqslant \frac{N(X_r)N(Y_r)}{1-N(Y_r)} \\ &\leqslant \frac{N(X_r)N(Y)^{2^r}}{1-N(Y)^{2^r}} \\ &\leqslant \frac{N(X)N(Y)^{2^r}}{1-N(Y)}. \end{aligned}$$

The existence of X_r^{-1} can be shown in the same way as in our second scheme.

If we had used the lefthand residual matrix, the obvious changes would follow as before and we also have $Y_rX_r=X_rY_r$.

We can generalize our approach in the sense that we can define a recursion such that the residual (error) matrix can be any power of the previous residual matrix. For example, to get third-order convergence, we take the first three terms of our basic series and repeat after each step. Thus

$$\begin{aligned} X_{s+1} &= X_s(I+Y_s+Y_s^2) \\ Y_{s+1} &= I-AX_{s+1} \end{aligned}$$

with $X_0=X$, $Y_0=Y=I-AX$ and $N(Y)<1$. The proofs follow exactly as in the second-order case and we have

$$Y_{s+1}=Y^{3^{s+1}}.$$

However, another matrix multiplication needed at each step and the possible loss of significant figures are factors to be considered in the practical application of this scheme.

For any order convergence, say t, we have

$$\begin{aligned} X_{s+1} &= X_s(I+Y_s+Y_s^2+\ldots+Y_s^{t-1}) \\ Y_{s+1} &= I-AX_{s+1}. \end{aligned}$$

The proof is straightforward by induction on t and we also have

$$Y_{s+1}=Y^{t^{s+1}}.$$

The error bounds follow directly as before:
If $N(Y) < 1$,

$$N(A^{-1}-X_s) \leq \frac{N(X_sY_s)}{1-N(Y_s)}$$

$$\leq \frac{N(X_s)N(Y_s)}{1-N(Y_s)}$$

$$\leq \frac{N(X_s)N(Y)^{t^s}}{1-N(Y)^{t^s}}$$

$$\leq \frac{N(X)N(Y)^{t^s}}{1-N(Y)},$$

the first being the best, but the last being useful in determining s as was indicated on page 266.

We note that the multiprecision needed for the effective use of these higher order convergence schemes might be a minimal problem in view of the hardware being developed by the computer industry. At the end of section 1 we gave an example using different computers and we found that the CDC 6400 gave extraordinary results.

Let us now discuss the relative error of the approximate inverse derivable from these schemes. For each scheme we can use

$$A^{-1}-X_k = A^{-1}(I-AX_k) = (I-X_kA)A^{-1}$$

so that

$$\frac{N(A^{-1}-X_k)}{N(A^{-1})} \leq N(Y_k),$$

where Y_k may represent the lefthand or the righthand residual matrix. Hence, for our first scheme:

$$\frac{N(A^{-1}-X_k)}{N(A^{-1})} \leq N(Y);$$

for the second:

$$\frac{N(A^{-1}-X_k)}{N(A^{-1})} \leq N(Y^{k+1})$$

$$\leq N(Y)^{k+1};$$

and for the third:

$$\frac{N(A^{-1}-X_r)}{N(A^{-1})} \leq N(Y^{2^r})$$

$$\leq N(Y)^{2^r}.$$

In general for tth order convergence:

$$\frac{N(A^{-1}-X_s)}{N(A^{-1})} \leq N(Y^{t^s})$$

$$\leq N(Y)^{t^s}.$$

Hence, if we use this relative error as a criterion for stopping the process, it is evident that the third is far better than the first two. There are also other considerations which will be mentioned shortly. To get a better bound we can use the relation

$$\frac{1}{N(A^{-1})} \leq \frac{1+N(Y_k)}{N(X_k)},$$

which follows exactly as in section 1. Hence, for our first scheme:

$$\frac{N(A^{-1}-X_k)}{N(A^{-1})} \leqslant \frac{N(XY^{k+1})}{N(X)} \cdot \frac{1+N(Y)}{1-N(Y)}$$

$$\leqslant N(Y)^{k+1} \cdot \frac{1+N(Y)}{1-N(Y)};$$

for the second scheme:

$$\frac{N(A^{-1}-X_k)}{N(A^{-1})} \leqslant \frac{N(X_kY_k)}{N(X_k)} \cdot \frac{1+N(Y_k)}{1-N(Y_k)}$$

$$\leqslant N(Y_k) \cdot \frac{1+N(Y_k)}{1-N(Y_k)}$$

$$\leqslant N(Y)^{k+1} \cdot \frac{1+N(Y)^{k+1}}{1-N(Y)^{k+1}};$$

and in general for the $t$the order convergence scheme:

$$\frac{N(A^{-1}-X_s)}{N(A^{-1})} \leqslant \frac{N(X_sY_s)}{N(X_s)} \cdot \frac{1+N(Y_s)}{1-N(Y_s)}$$

$$\leqslant N(Y_s) \cdot \frac{1+N(Y_s)}{1-N(Y_s)}$$

$$\leqslant N(Y)^{t^s} \cdot \frac{1+N(Y)^{t^s}}{1-N(Y)^{t^s}}.$$

Let us compare the three main schemes we have discussed. Our basis for comparison will be the number of new matrix multiplications needed to advance one step in the process and get the error bound for that particular iterate, presuming the previous one was not small enough. This means, of course, that the error bound must be calculated at each step. We have written the third scheme as $X_k+X_kY_k$ instead of $X_k(I+Y_k)$ since the product X_kY_k was calculated in the previous step for the evaluation of the error bound and is available. We will identify the first, second, and third schemes by Scheme I, Scheme II, and Scheme III in the following table and list the quantities that must be calculated in going from one step to the next.

	Scheme I		Scheme II		Scheme III	
	Step k	Step $k+1$	Step k	Step $k+1$	Step k	Step $k+1$
Iterate	X_k	X_k+XY^{k+1}	X_k	X_k+XY^{k+1}	X_k	$X_k+X_kY_k$
Residual	Y	Y	Y_k	Y_kY	Y_k	Y_kY_k
Bound for $N(A^{-1}-X_k)$...	$\frac{N(XY^{k+1})}{1-N(Y)}$	$\frac{N(XY^{k+2})}{1-N(Y)}$	$\frac{N(X_kY_k)}{1-N(Y_k)}$	$\frac{N(X_{k+1}Y_{k+1})}{1-N(Y_{k+1})}$	$\frac{N(X_kY_k)}{1-N(Y_k)}$	$\frac{N(X_{k+1}Y_{k+1})}{1-N(Y_{k+1})}$
Additional products.....	1 XY^{k+2}		3 XY^{k+1}, Y_kY, $X_{k+1}Y_{k+1}$		2 Y_kY_k, $X_{k+1}Y_{k+1}$	

We see that the second scheme has one more matrix multiplication than the third and, since the latter has a much better rate of convergence, the third scheme is definitely better than the second. In comparing the first and the third, the question of the additional matrix multiplication to get better convergence arises and cannot be answered in general. The exponent of Y in Scheme I may be such as to demand multiprecision hardware and this may be worse than performing an

extra matrix multiplication. In some cases too, this extra multiplication may itself demand multiprecision hardware. In view of the quadratic convergence, though, many fewer steps should be required for the third scheme and probably multiprecision hardware would not be needed (beyond double precision, that is). We note that in our second scheme, the convergence is better than in the first and we have proven that the bounds are also better. *Two* additional matrix multiplications may not be worth it, however. Considering all of these factors and also keeping in mind the relative error, we feel that the third scheme is the most practical.

We would like to show now that two frequently used iteration procedures to find the inverse of a matrix are actually included in our development. The first is defined by the recurrence

$$X_{k+1}=X+(I-XA)X_k, \qquad k\geq 0, \tag{A}$$

with $X_0=0$. Letting $Y_k=I-AX_k$, we can rewrite this recurrence as

$$\begin{aligned} X_{k+1}&=X+X_k-XAX_k\\ &=X_k+X(I-AX_k)\\ &=X_k+XY_k. \end{aligned}$$

Now

$$\begin{aligned} Y_k&=I-AX_k\\ &=I-A(X_{k-1}+XY_{k-1})\\ &=I-AX_{k-1}-AXY_{k-1}\\ &=(I-AX)Y_{k-1}\\ &=YY_{k-1} \end{aligned}$$

and by induction we have

$$Y_k=Y^k.$$

So this recurrence can be written

$$\begin{aligned} X_{k+1}&=X_k+XY^k, \qquad k\geq 0\\ Y_{k+1}&=I-AX_{k+1}, \end{aligned}$$

which is exactly our second scheme (the starting value is the only difference). However, in the original form, only the lefthand residual can be used whereas in our development either one is applicable. We note also that for the convergence of (A) each eigenvalue of the righthand residual must be of modulus less than one. Although this seems to imply the necessity of working with both residuals, we realize that this is not the case since the two residuals are similar:

$$I-AX=A(I-XA)A^{-1}.$$

The second frequently used iteration is defined by

$$X_{k+1}=X_k(2I-AX_k), \qquad k\geq 0 \tag{B}$$

where X_0 is arbitrary. The immediate difficulty is the choice of X_0 and hence this iteration is more useful as an improvement scheme. If we let $X_0=X$ and $Y_k=I-AX_k$, we can rewrite (B) as

$$X_{k+1}=X_k(I+Y_k)$$

which is identical to our third scheme. The changes to make use of the lefthand residual, once (B) is written this way, follow exactly as in our development.

So we see that both of these iteration formulas can be thought of as having their foundation in our basic relation that

$$A^{-1}=X+XY+XY^2+\ldots.$$

We have repeatedly mentioned that in our schemes we can improve either the lefthand or the righthand inverse. This is a very important consideration in the solution of $Ax=b$, as will be pointed out in the next section.

For a numerical example, let A be the 6×6 segment of the Hilbert Matrix and to generate the approximate inverse, X, we use the same program and computer as in section 1, except that we used no iterative improvement scheme. We used double precision accumulation for the products XY and our criterion for stopping was when the elements of XY^k were in magnitude less than 10^{-8}. The final results for each scheme were identical.

For convenience we list the results separately and then make some comments about them.

Scheme I

$$N(A^{-1}-X)\leq N(XY+XY^2+\ldots+XY^5) \qquad =.412\,E+05$$

$$N(A^{-1}-X_5)\leq\frac{N(XY^6)}{1-N(Y)} \qquad =.227\,E-07$$

$$\leq\frac{N(X)N(Y)^6}{1-N(Y)} \qquad =.199\,E-02$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})}\leq N(Y) \qquad =.234\,E-01$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})}\leq N(XY+XY^2+\ldots+XY^5)*\frac{1+N(Y)}{N(X)} \qquad =.360\,E-02$$

$$\frac{N(A^{-1}-X_5)}{N(A^{-1})}\leq\frac{N(XY^6)}{N(X)}\,\frac{1+N(Y)}{1-N(Y)} \qquad =.191\,E-14$$

$$\leq N(Y)^6\frac{1+N(Y)}{1-N(Y)} \qquad =.170\,E-09$$

and for

$$\frac{N(X_5)}{1+N(Y)}\leq N(A^{-1})\leq\frac{N(X_5)}{1-N(Y)},$$

$$.117\,E+08\leq N(A^{-1})\leq .122\,E+08$$

Scheme II

$$N(A^{-1}-X_5) \leq \frac{N(X_5Y_5)}{1-N(Y_5)} = .114\ E-04$$

$$\leq \frac{N(X_5)N(Y_5)}{1-N(Y_5)} = .502\ E-04$$

$$\leq \frac{N(X)N(Y)^6}{1-N(Y)^6} = .194\ E-02$$

$$\frac{N(A^{-1}-X_5)}{N(A^{-1})} \leq N(Y_5) = .421\ E-11$$

$$\frac{N(A^{-1}-X_5)}{N(A^{-1})} \leq \frac{N(X_5Y_5)}{N(X_5)} \cdot \frac{1+N(Y_5)}{1-N(Y_5)} = .952\ E-12$$

$$\leq N(Y)^6 \cdot \frac{1+N(Y)^6}{1-N(Y)^6} = .162\ E-09$$

and for

$$\frac{N(X_5)}{1+N(Y_5)} \leq N(A^{-1}) \leq \frac{N(X_5)}{1-N(Y_5)},$$

$$.11931731\ E+08 \leq N(A^{-1}) \leq .11931731\ E+08$$

Scheme III

$$N(A^{-1}-X_3) \leq \frac{N(X_3Y_3)}{1-N(Y_3)} = .840\ E-05$$

$$\leq \frac{N(X_3)N(Y_3)}{1-N(Y_3)} = .505\ E-04$$

$$\leq \frac{N(X)N(Y)^8}{1-N(Y)} = .108\ E-05$$

$$\frac{N(A^{-1}-X_3)}{N(A^{-1})} \leq N(Y_3) = .424\ E-11$$

$$\frac{N(A^{-1}-X_3)}{N(A^{-1})} \leq \frac{N(X_3Y_3)}{N(X_3)} \cdot \frac{1+N(Y_3)}{1-N(Y_3)} = .704\ E-12$$

$$\leq N(Y)^8 \cdot \frac{1+N(Y)^8}{1-N(Y)^8} = .886\ E-13$$

and

$$\frac{N(X_3)}{1+N(Y_3)} \leq N(A^{-1}) \leq \frac{N(X_3)}{1-N(Y_3)},$$

$$.11931731\ E+08 \leq N(A^{-1}) \leq .11931731\ E+08.$$

We would like to make some comments about this example. One of our theoretical results was that

$$N(Y_k) \leqslant N(Y)^{k+1}.$$

Now $N(Y)^6 \approx .162\ E-09$, so that this inequality does actually hold.

Another inequality we proved was that for Scheme II

$$\frac{N(X_kY_k)}{1-N(Y_k)} \leqslant \frac{N(XY^{k+1})}{1-N(Y)}.$$

We notice here, however, that this is not the case for $k=5$. In investigating further we found that this inequality did hold for $k=1, 2, 3$, and for $k=4$ there was almost equality. In examining the significant digits involved, we found that after $k=3$, at which point $N(Y_3) \approx .1\ E-08$, we had gone beyond the accuracy of the numbers with which we were dealing. We were working in double precision with numbers originally of the order $E+08$ and have now gone beyond 16 digits. Hence, any information beyond this point becomes meaningless.

The difficulty is more apparent in dealing with X_kY_k and XY^{k+1} than with just Y_k and Y because of the matrix multiplications involved which lead to more serious round off errors. This is the reason that we mentioned in this chapter the possible need for multiprecision when working with these schemes.

The contradiction in Scheme III is different but comes from the same source. Here

$$\frac{N(X_3Y_3)}{1-N(Y_3)} > \frac{N(X)N(Y)^8}{1-N(Y)}$$

and $k=3$, the quadratic convergence making the difficulty show up earlier.

This example very clearly illustrates the point of the question asked in the Introduction: How do we know these numbers *are* the solution of our problem? The bounds developed here certainly help in answering this question.

4. Improvement Schemes for the Approximate Solution Vector of $Ax=b$

Let us now examine ways of improving the solution vector, z, of a linear algebraic system, $Az=b$. As in the previous section, the residual, $r=b-Az$, plays an important role. The standard approach is to solve the system $Ax=r$ whose solution, d, is an improvement to be added to z. The usual convergence proof depends on Wilkinson's backward analysis which says that z is the exact solution of a slightly perturbed system $(A+E)x=b$. The condition for convergence of the iterative improvement scheme [12] is that $N(A^{-1}E) < \frac{1}{2}$. We would like to present a slightly different scheme and give two convergence proofs for it. We will then give a different convergence proof for the standard approach, one we find more meaningful.

The alternate scheme differs in that the residual that is used comes from the system just solved and not from the original one; that is, $r_0=b-Az$ and if d_0 is the solution of $Ax=r_0$, then $r_1=r_0-Ad_0$. In general we have

$$r_{k+1}=r_k-Ad_k$$

and d_k is the approximate solution of $Ax=r_k$.

THEOREM 4.1. *Let the vectors* r_k *and* d_k *be defined as above. Then if* $N(r_{k+1}) < aN(r_k), 0 < a < 1$, r_k *approaches zero as* k *goes to infinity and* $z+d_0+d_1+\ \ldots\ +d_k$ *converges to* $A^{-1}b$.

PROOF: We have

$$r_0 = b - Az$$

$$r_1 = r_0 - Ad_0$$

$$r_2 = r_1 - Ad_1$$

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

$$r_{k+1} = r_k - Ad_k$$

so that

$$r_0 + r_1 + \cdots + r_{k+1} = b + r_0 + r_1 + \cdots + r_k - A(z + d_0 + d_1 + \cdots + d_k)$$

or

$$r_{k+1} = b - A(z + d_0 + d_1 + \cdots + d_k).$$

Now

$$N(r_{k+1}) < aN(r_k) < a^2N(r_{k-1}) < \cdots < a^{k+1}N(r_0).$$

Hence, as k increases to infinity, a^{k+1} goes to zero, and then $N(r_{k+1})$ goes to zero which implies that r_{k+1} goes to zero. Therefore $A(z + d_0 + d_1 + \cdots + d_k)$ approaches b which means that $z + d_0 + d_1 + \cdots + d_k$ converges to $A^{-1}b$.

Instead of summing the residuals, if we use Wilkinson's backward approach, we could say that d_k is the exact solution of a slightly perturbed system and we would have

$$(A + E_k)d_k = r_k$$

where we have subscripted the E to show the dependence on each system. With the introduction of E_k we have another convergence criterion.

THEOREM 4.2. *Let the vectors* r_k, d_k *and the matrix* E_k *be defined as above. Assume* $N(E_k) < c$ *for all* k. *Then if* $N(d_{k+1}) < pN(d_k)$, $0 < p < 1$, r_k *approaches zero as* k *goes to infinity and* $z + d_0 + d_1 + \cdots + d_k$ *converges to* $A^{-1}b$.

PROOF:

$$(A + E_k)d_k = r_k$$

$$Ad_k + E_kd_k = r_k$$

$$E_kd_k = r_k - Ad_k$$

but

$$r_{k+1} = r_k - Ad_k$$

so that

$$r_{k+1} = E_kd_k$$

and

$$N(r_{k+1}) \leq N(E_k)N(d_k).$$

Now

$$N(d_k) < pN(d_{k-1}) < p^2N(d_{k-2}) < \cdots < p^kN(d_0).$$

Hence,

$$N(r_{k+1}) < cp^kN(d_0)$$

so that as k goes to infinity, p^k approaches zero and r_{k+1} approaches zero and the conclusion follows just as in Theorem 4.1.

COROLLARY. *Under the same conditions as the theorem, we have* $N[A^{-1}b-(z+d_0+\ldots+d_k)]\leq\frac{p^{k+1}}{1-p}N(d_0)$.

PROOF:

$$A^{-1}b-(z+d_0+\ldots+d_k)=d_{k+1}+d_{k+2}+\ldots$$

$$N[A^{-1}b-(z+d_0+\ldots+d_k)]\leq N(d_{k+1})+N(d_{k+2})+\ldots$$

$$\leq p^{k+1}N(d_0)+p^{k+2}N(d_0)+\ldots$$

$$=p^{k+1}(1+p+p^2+\ldots)N(d_0)$$

$$=\frac{p^{k+1}}{1-p}N(d_0).$$

We will see that these two conditions are actually contained in our next theorem. There $N(I-XA)<1$ is our condition, and this is the p here; and also $N(I-AX)<1$ is used, and this is the a here.

It would seem, however, that repeatedly calculating the residuals from the primary equation, $b-Ax$, is a better scheme since the magnitude of the residuals that are being used may be much smaller for the alternate method. Using the smaller numbers might demand multiprecision routines, more than the double precision necessary for the calculation of each residual in the standard approach. On the other hand, it has been found that $N(r_{k+1})$ is not always smaller than $N(r_k)$ for the standard scheme, and in similar problems using the other scheme we have found that the residuals do decrease at each step. This could be due to the fact that in the second scheme we make one addition at the end, $z+d_0+d_1+\ldots+d_k$, for our final answer and thus only one rounding error occurs. In the standard scheme, however, this is done at each step and if this rounding error is included in the convergence proof, we would find that the error, $A^{-1}b-x_k$, does not approach zero but only gets small [12, p. 110]. Hence, this convergence proof is really for what Forsythe calls an "Almost Theorem" [13, p. 9], especially since Kahan has found a counterexample to the standard approach [10, p. 78]. In practice we have found both schemes produce identical results.

Let us now turn our attention to a different convergence proof for the standard scheme. It will be useful to write out a few steps of the process. Let d_k be the computed solution at each step.

(1) solve $Ax=b$:

$$z=Xb$$

$$r=b-Az$$

$$=b-A(Xb)$$

$$=(I-AX)b$$

(2) solve $Ax=r$:

$$d_0=Xr$$

$$=X(b-Az)$$

$$=Xb-XAz$$

$$=(I-XA)z$$

$$=Yz,\ Y=I-XA.$$

improved solution:

$$z_1 = z + d_0$$
$$= z + Yz$$
$$= (I+Y)z$$

and

$$r_1 = b - Az_1$$
$$= b - A(z + d_0)$$
$$= b - Az - Ad_0$$
$$= r - AXr$$
$$= (I - AX)r.$$

(3) solve $Ax = r_1$:

$$d_1 = Xr_1$$
$$= X(I - AX)r$$
$$= (I - XA)Xr$$
$$= YXr$$
$$= Y^2 z$$

improved solution:

$$z_2 = z_1 + d_1$$
$$= (I+Y)z + Y^2 z$$
$$= (I + Y + Y^2)z$$

and

$$r_2 = b - Az_2$$
$$= b - A(z_1 + d_1)$$
$$= b - Az_1 - Ad_1$$
$$= r_1 - AXr_1$$
$$= (I - AX)r_1$$
$$= (I - AX)^2 r.$$

In general, we have

$$d_{k-1} = Xr_{k-1}$$
$$= Yd_{k-2}$$
$$= Y^k z,$$

$$z_k = z_{k-1} + d_{k-1}$$
$$= (I + Y + Y^2 + \ldots + Y^k)z,$$

$$r_k = (I - AX)r_{k-1}$$
$$= (I - AX)^k r$$

406-680 O - 71 - 4

for by induction

$$\begin{aligned} d_k &= Xr_k \\ &= X(I-AX)^k r \\ &= X(I-AX)(I-AX)^{k-1} r \\ &= YXr_{k-1} \\ &= Yd_{k-1} \\ &= Y^{k+1} z; \end{aligned}$$

$$\begin{aligned} z_{k+1} &= z_k + d_k \\ &= (I+Y+Y^2+\ldots+Y^k)z + Y^{k+1}z \\ &= (I+Y+Y^2+\ldots+Y^k+Y^{k+1})z; \end{aligned}$$

$$\begin{aligned} r_{k+1} &= b - Az_{k+1} \\ &= b - A(z_k + d_k) \\ &= b - Az_k - Ad_k \\ &= r_k - AXr_k \\ &= (I-AX)r_k \\ &= (I-AX)^{k+1} r. \end{aligned}$$

THEOREM 4.3. *Let the vectors* d_k, z_k, r_k *and the matrix* Y *be defined as above. Then if* $N(Y) < 1$, z_k *converges to* $A^{-1}b$ *as* k *goes to infinity.*

PROOF: $z_k = (I+Y+Y^2+\ldots+Y^k)z$ and, as we proved in Lemma 1.2, if $N(Y) < 1$, $I+Y+Y^2+\ldots+Y^k$ converges to $(I-Y)^{-1}$ as k goes to infinity. Also

$$Y = I - XA$$

$$XA = I - Y$$

$$A^{-1}X^{-1} = (I-Y)^{-1}.$$

Hence, as k goes to infinity, z_k converges to

$$(I-Y)^{-1}z = (A^{-1}X^{-1})(Xb) = A^{-1}b.$$

COROLLARY 1. *Let* d_k *and* r_k *be defined as above. Then if* $N(Y) < 1$, d_k *goes to zero as* k *goes to infinity and if* $N(I-AX) < 1$, r_k *also goes to zero.*

PROOF:

$$d_{k-1} = Y^k z$$

so that

$$N(d_{k-1}) \leq N(Y)^k N(z).$$

Since $N(Y) < 1$, the righthand side goes to zero as k goes to infinity so that the lefthand side does also, which implies that d_{k-1} goes to zero. Since

$$r_k = (I - AX)^k r$$

and

$$N(r_k) \leq N(I - AX)^k N(r),$$

if $N(I - AX) < 1$, the same argument implies that r_k goes to zero as k goes to infinity.

We still have linear convergence but we feel using $N(Y) < 1$ is more meaningful than $N(A^{-1}E) < \frac{1}{2}$. Since we need X if we are going to use any bounds for $N(A^{-1}b - z_k)$, it is also a practical criterion.

From this theorem we can also get an estimate of the improvement at each step as well as a final error bound.

COROLLARY 2. *With the same definitions and conditions as in the theorem,*

$$(a) \quad N(z_k - z_{k-1}) \leq N(Y)^k N(z) \text{ and}$$

$$(b) \quad N(A^{-1}b - z_k) \leq \frac{N(Y)^{k+1}}{1 - N(Y)} N(z)$$

PROOF:

$$(a) \quad z_k - z_{k-1} = d_{k-1}$$
$$= Y^k z$$

so that

$$N(z_k - z_{k-1}) \leq N(Y)^k N(z).$$

$$(b) \quad A^{-1}b - z_k = (I - Y)^{-1} z - z_k$$
$$= (I - Y)^{-1} z - (I + Y + \ldots + Y^k) z$$
$$= (Y^{k+1} + Y^{k+2} + \ldots) z$$

so that

$$N(A^{-1}b - z_k) \leq \frac{N(Y^{k+1})}{1 - N(Y)} N(z)$$
$$\leq \frac{N(Y)^{k+1}}{1 - N(Y)} N(z).$$

This is a smaller number generally than that bounding the difference between two successive steps in the iteration. In fact, if $N(Y) \leq \frac{1}{2}$,

$$\frac{N(Y)}{1 - N(Y)} \leq 1$$

and

$$N(A^{-1}b - z_k) \leq \frac{N(Y)^{k+1}}{1 - N(Y)} N(z)$$
$$= N(Y)^k N(z) \cdot \frac{N(Y)}{1 - N(Y)}$$
$$\leq N(Y)^k N(z).$$

We note that for the convergence of the sequence of z_k we need only that the norm of the lefthand residual be less than one. This implies that the residuals, $b - Az_k$, need not go to zero as k

increases and this has been the case in some problems as previously mentioned. This fact also emphasizes that in the solution of $Ax=b$, the lefthand inverse is the more important one. As was pointed out before, a good right inverse may not imply that the error in the solution vector is small. Hence, if we intend to use an approximate inverse in our solution of $Ax=b$ it is very important that the improvement scheme used be one that improves the lefthand inverse. This is very easily done by the methods of the previous section.

5. Error Bounds for the Approximate Inverse of Special Matrices

In this section we will briefly discuss some results concerning a symmetric matrix, derive an error bound for the inverse of a triangular matrix, and then investigate the LU decomposition of a matrix A for possible error bounds on A^{-1} which we will apply to the solution of $Ax=b$.

Since we will be developing results that apply to some special matrices, it might be good to give some properties of these matrices so that they will be readily available. We will not give any proofs as these can be found in various texts on linear algebra or numerical linear algebra as listed in the bibliography.

Let A be a real symmetric matrix.

1. The eigenvalues of A^2 are the squares of the eigenvalues of A.
2. The eigenvalue of maximum modulus of A^*A is the square of that of A. (A^* is the conjugate transpose of A and hence, in our case, is just the transpose of A, A^T.)
3. The spectral norm of any matrix A, $N_s(A)$, is defined as the square root of the eigenvalue of maximum modulus of A^*A. Hence, in our case, $N_s(A)$ is just the absolute value of the eigenvalue of A which has maximum modulus.
4. A is orthogonally similar to a diagonal matrix; i.e., $A=O^TDO$ where O is orthogonal, $O^T=O^{-1}$ and D is diagonal.

We will also use a few general relations which we will mention at this time.

1. Let $N_f(A)$ be the Frobenius norm of A. Then if A and B are orthogonally similar, $N_f(A)=N_f(B)$. (See section 1, where we proved this for unitarily similar matrices.)
2. We recall a theorem of Schur (also used in sect. 1): If A is an arbitrary matrix, then it can be transformed to a triangular matrix by means of a unitary matrix; i.e., a unitary matrix U exists such that $U^*AU=T$, where $U^*=U^{-1}$ and T is upper triangular.

We will now define a relation between matrices which may not be too familiar.

DEFINITION. A *is dominated by* B *if and only if* $|a_{ij}| \leq b_{ij}$ *for all* i *and* j. *Symbolically, we write* $A \ll B$. *Some basic properties of this relation are that if* $A_1 \ll B_1$ *and* $A_2 \ll B_2$, *then* $A_1+A_2 \ll B_1+B_2$ *and* $A_1A_2 \ll B_1B_2$.

Finally, we will use the symbol J to represent the $n \times n$ matrix whose elements are all ones, and l_i will be used for the eigenvalues of the matrix under consideration.

With this background we can proceed to our first theorem.

THEOREM 5.1. *If* A *is a real symmetric matrix and is* $0 < r \leq |l_i|$, *then* $N_s(A^{-1}) \leq 1/r$.

PROOF:

$$r \leq |l_i|$$

$$\frac{1}{|l_i|} \leq \frac{1}{r}$$

$$|l_i(A^{-1})| \leq 1/r$$

$$\max |l_i(A^{-1})| \leq 1/r$$

and

$$N_s(A^{-1}) \leq 1/r.$$

In our next few theorems let the following notation hold. Let X be the approximate inverse of A and let $R=OXO^T=(r_{ij})$ where O is an orthogonal matrix such that $A=O^TDO$.

THEOREM 5.2. *Let* A *be a real symmetric matrix. Then if*

$$N_f(I - AX) \leq e, \quad N_f(A^{-1} - X) \leq n^{1/2} \frac{e}{\min |l_i|}.$$

PROOF:

$$A^{-1} - X = O^T D^{-1} O - O^T R O$$

$$= O^T (D^{-1} - R) O$$

and

$$N_f(A^{-1} - X) = N(D^{-1} - R)$$

$$= N[D^{-1}(I - DR)].$$

Now

$$I - AX = I - O^T D R O$$

$$= O^T (I - DR) O$$

so that

$$N_f(I - AX) = N_f(I - DR).$$

Hence

$$N_f(A^{-1} - X) \leq N_f(D^{-1}) N_f(I - AX).$$

Now

$$D^{-1} \ll \frac{1}{\min |l_i|} I$$

so that

$$N_f(D^{-1}) \leq \frac{n^{1/2}}{\min |l_i|}$$

and if $N_f(I - AX) \leq e$, we have

$$N_f(A^{-1} - X) \leq n^{1/2} \frac{e}{\min |l_i|}.$$

COROLLARY 1. *Under the same conditions as the theorem except that if* $I - AX \ll kJ$, *we have*

$$N_f(A^{-1} - X) \leq n^{3/2} \frac{k}{\min |l_i|}.$$

PROOF: Since $N_f(J) = n$, we use nk instead of e as in the theorem.

COROLLARY 2. *Let all the eigenvalues of* A *be the same, say a. Then*

$$N_f(A^{-1} - X) \leq \frac{e}{|a|}.$$

PROOF: We use the fact that $D = aI$ and the property of a norm.

THEOREM 5.3. *Let the* n *eigenvalues of* A *be distinct. Let* $E = (I - XA) - (I - AX)$ *and let* $E_1 = OEO^T = (e_{ij})$. *Then if* $d_i = l_i^{-1} - r_{ii}$,

$$N_f(A^{-1} - X) \leq n \max \left(|d_i|, \left| \frac{e_{ij}}{l_i - l_j} \right| \right).$$

PROOF:

$$E = (I - XA) - (I - AX)$$
$$= AX - XA$$
$$= O^T(DR - RD)O.$$

If

$$E_1 = OEO^T, \qquad E_1 = DR - RD.$$

In terms of elements, this last equation can be written

$$e_{ij} = l_i r_{ij} - l_j r_{ij}, \quad i \neq j$$
$$= 0 \qquad , \quad i = j$$

or

$$r_{ij} = \frac{e_{ij}}{l_i - l_j} \qquad , \quad i \neq j$$

and for $i = j$, r_{ij} is not determined but they should be close to l_i^{-1}. Now consider $D^{-1} - R$. The diagonal elements are d_i and the off-diagonal, $-e_{ij}/(l_i - l_j)$. Then

$$D^{-1} - R \ll \max\left(|d_i|, \left|\frac{e_{ij}}{l_i - l_j}\right|\right) J$$

so that

$$N_f(A^{-1} - X) = N_f(D^{-1} - R)$$
$$\leq n \max\left(|d_i|, \left|\frac{e_{ij}}{l_i - l_j}\right|\right).$$

The spread of the eigenvalues of a matrix can give a good idea of the condition of a matrix; in fact, the ratio of the modulus of the largest to that of the smallest is defined as the P-condition of a matrix. We see from our error bound that a large spread does not necessarily mean we cannot get a good approximate inverse. This bound implies, however, that if two eigenvalues are very close, we might expect trouble but that this might be counteracted if E is small enough. Hence, once again knowledge of the lefthand residual matrix is useful.

COROLLARY. *IF* $\mathrm{E} = 0$, $\mathrm{N_f(A^{-1} - X)} \leq \mathrm{n} \max_{\mathrm{i}} |\mathrm{d_i}|$.

PROOF: If $E = 0$, $E_1 = 0$ and then $r_{ij} = 0$ for $i \neq j$. Hence

$$D^{-1} - R = \text{diagonal } (d_i)$$

$$\ll \max |d_i| J$$

so that

$$N_f(A^{-1} - X) \leq n \max |d_i|.$$

Let us now consider another special case, a triangular matrix T, and derive an error bound for its inverse. Let N_{rs} be the maximum row sum norm.

THEOREM 5.4. *Let* T *be a triangular matrix and let* $\mathrm{e} = \max_{\mathrm{i}} \left|\frac{\mathrm{t_{ij}}}{\mathrm{t_{ii}}}\right|$ *for all* j. *Then*

$$\mathrm{N_{rs}(T^{-1})} \leq \frac{(1 + \mathrm{e})^{\mathrm{n}-1}}{\min |\mathrm{t_{ii}}|}.$$

PROOF: Let $T=D-U=D(I-D^{-1}U)=D(I-E)$, D diagonal, U strictly upper and $E=D^{-1}U=(e_{ij})$ where $e_{ij}=t_{ij}/t_{ii}$. Then $T^{-1}=(I-E)^{-1}D^{-1}$. Let us examine $(I-E)^{-1}$. Since E has zero diagonal, $E^n=0$ and $(I-E)^{-1}=I+E+E^2+\cdots+E^{n-1}$. Let $e=\max_i |e_{ij}|$, for all j. Then we can write $E \ll eP$ where

$$P=\begin{bmatrix} 0 & 1 & 1 & 1 & \cdots & 1 \\ & 0 & 1 & 1 & \cdots & 1 \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \vdots \\ & & & & \ddots & \vdots \\ & & & & 0 & 1 \\ & & & & & 0 \end{bmatrix}.$$

Let us examine the various powers of P.

$$P^2=\begin{bmatrix} 0 & 0 & 1 & 2 & 3 & 4 & \cdots & n-2 \\ & 0 & 0 & 1 & 2 & 3 & \cdots & n-3 \\ & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \ddots & \ddots & 2 \\ & & & & & & \ddots & 1 \\ & & & & & & & 0 \\ & & & & & & & 0 \end{bmatrix}$$

$$P^3=\begin{bmatrix} 0 & 0 & 0 & 1 & 3 & 6 & 10 & 15 & \cdots & \binom{n-2}{2} \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 10 \\ & & & & & \ddots & \ddots & \ddots & \ddots & 6 \\ & & & & & & \ddots & \ddots & \ddots & 3 \\ & & & & & & & \ddots & \ddots & 1 \\ & & & & & & & & \ddots & 0 \\ & & & & & & & & & 0 \\ & & & & & & & & & 0 \end{bmatrix}$$

In P^3 we have written $\binom{n-2}{2}$ instead of $1+2+3+\ldots+n-3$ which comes from multiplying the first row of P times the nth column of P^2. We also note that $n-2$ can be written $\binom{n-2}{1}$ which we will use in P^2; of course $\binom{n-2}{0}=1$, the element in the $(1, n)$ position of P^1. Generalizing this pattern and realizing that $P^n=0$, we find that P^{n-1} has all zeros except for a 1 in the $(1, n)$ position which we will write as $\binom{n-2}{n-2}$. Let us write the first row (only) of the next few powers:

$$P_1^4=\left[0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 4 \quad 10 \quad 20 \quad \ldots \quad \binom{n-2}{3}\right]$$

$$P_1^5=\left[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 5 \quad 15 \quad \ldots \quad \binom{n-2}{4}\right].$$

We restrict ourselves to the maximum row sum norm and our interest is in

$$N(I+E+E^2+\ldots+E^{n-1}).$$

As mentioned in our first chapter, it will be better to add the matrices first and then take the norm. It is clear that the maximum row sum will come from the first row. Let us rewrite the first rows of these matrices, then, using the binomial expression $\binom{n}{k}$.

$$P_1^1=\left[0 \quad \binom{0}{0} \quad \binom{1}{0} \quad \binom{2}{0} \quad \binom{3}{0} \quad \binom{4}{0} \quad \binom{5}{0} \quad \ldots \quad \binom{n-2}{0}\right]$$

$$P_1^2=\left[0 \quad 0 \quad \binom{1}{1} \quad \binom{2}{1} \quad \binom{3}{1} \quad \binom{4}{1} \quad \binom{5}{1} \quad \ldots \quad \binom{n-2}{1}\right]$$

$$P_1^3=\left[0 \quad 0 \quad 0 \quad \binom{2}{2} \quad \binom{3}{2} \quad \binom{4}{2} \quad \binom{5}{2} \quad \ldots \quad \binom{n-2}{2}\right]$$

$$P_1^4=\left[0 \quad 0 \quad 0 \quad 0 \quad \binom{3}{3} \quad \binom{4}{3} \quad \binom{5}{3} \quad \ldots \quad \binom{n-2}{3}\right]$$

We have $E \ll eP$ so that $E^k \ll e^k P^k$. Let us take the sum of the various powers position by position; i.e., find the sum for the (1, 1) position, (1, 2) position, . . ., $(1, n)$ position.

$$(1,1)\colon 1 \qquad = 1$$

$$(1,2)\colon \binom{0}{0} e \qquad = e(1+e)^0$$

$$(1,3)\colon \binom{1}{0} e+\binom{1}{1} e^2 \qquad = e(1+e)^1$$

$$(1,4)\colon \binom{2}{0} e+\binom{2}{1} e^2+\binom{2}{2} e^3 \qquad = e(1+e)^2$$

$$(1,5)\colon \binom{3}{0} e+\binom{3}{1} e^2+\binom{3}{2} e^3+\binom{3}{3} e^4 \qquad = e(1+e)^3$$

$$\ldots$$

$$(1,n)\colon \binom{n-2}{0} e+\binom{n-2}{1} e^2+\binom{n-2}{2} e^3+\ldots+\binom{n-2}{n-2} e^{n-1} = e(1+e)^{n-2}$$

The norm, therefore, comes from adding the extreme righthand column:

$$\begin{aligned} N_{rs}[(I-E)^{-1}] &= 1 + e(1+e)^0 + e(1+e)^1 + e(1+e)^2 + \ldots + e(1+e)^{n-2} \\ &= 1 + e[1 + (1+e) + (1+e)^2 + \ldots + (1+e)^{n-2}] \\ &= 1 + e\left[\frac{(1+e)^{n-1} - 1}{e}\right] \\ &= (1+e)^{n-1}. \end{aligned}$$

Hence we have

$$\begin{aligned} N_{rs}(T^{-1}) &\leq N_{rs}(D^{-1})(1+e)^{n-1} \\ &= \frac{(1+e)^{n-1}}{\min\limits_i |t_{ii}|} \end{aligned}$$

and all the quantities are readily available by inspection of T.

COROLLARY 1.

$$N_{rs}(T^{-1}) \leq \frac{\left(\min\limits_i |t_{ii}| + \max\limits_i |t_{ij}|\right)^{n-1}}{\min\limits_i |t_{ii}|^n}.$$

PROOF: We use the fact that

$$\max_i \left|\frac{t_{ij}}{t_{ii}}\right| \leq \frac{\max\limits_i |t_{ij}|}{\min\limits_i |t_{ii}|} \quad \text{for all } j$$

and substitute this inequality in the bound of the theorem.

We see in this last bound that the two important numbers are the smallest and the largest. This ties in very well with the concept of the condition of a matrix, because a matrix with a large spread in the magnitude of its elements is generally considered an ill-conditioned matrix. The larger the difference of the maximum element and the minimum element, the higher the value of this bound.

COROLLARY 2.

$$\textit{If } e \leq 1, \quad N_{rs}(T^{-1}) \leq \frac{2^{n-1}}{\min |t_{ii}|}.$$

This corollary would definitely be applicable to a diagonally dominant triangular matrix.

THEOREM 5.5. *Let* A *be any matrix and let* e *be defined as in Theorem 5.4. Then if* $N_{rs}(I - AX) \leq q$,

$$N_f(A^{-1} - X) \leq nq \frac{(1+e)^{n-1}}{\min |l_i|}.$$

PROOF: If U is a unitary matrix and T triangular, we have

$$\begin{aligned} A &= U^*TU \\ X &= U^*RU, \qquad \text{for some } R \\ A^{-1} - X &= U^*(T^{-1} - R)U \end{aligned}$$

so that

$$\begin{aligned} N_f(A^{-1}-X) &= N_f(T^{-1}-R) \\ &= N_f(T^{-1}(I-TR)) \\ &\leq N_f(T^{-1})N_f(I-TR) \\ &= N_f(T^{-1})N_f(I-AX) \end{aligned}$$

following the same arguments that were used in the proof of Theorem 5.2.

Now for any A we have the relation $N_f(A) \leq n^{1/2}N_{rs}(A)$. Using this inequality we continue

$$N_f(A^{-1}-X) \leq nN_{rs}(T^{-1})N_{rs}(I-AX).$$

Using our bounds for $N_{rs}(T^{-1})$ we have

$$N_f(A^{-1}-X) \leq n\,\frac{(1+e)^{n-1}}{\min|l_i|}\,N_{rs}(I-AX)$$

where $e=\max_i \left|\frac{t_{ij}}{t_{ii}}\right|$ for all j, and l_i are the eigenvalues of A. The difficulty here, of course, is our lack of knowledge of t_{ij} but the bound certainly brings out the fact that the smaller the eigenvalue of A, the higher the bound. This also implies that for a nearly singular matrix (a zero eigenvalue), the bounds are much higher which gives another indication of the influence of ill conditioning on the calculation of the inverse.

A more practical application of the bounds for the inverse of a triangular matrix might be made to the error in the solution, z, of a linear algebraic system, $Ax=b$. We pointed out in section 2 that to get a bound on the error, $A^{-1}b-z$, we must use an approximate inverse. This necessitates solving an additional n systems. However, if we use the LU decomposition method (Gaussian Elimination), we need not do all that. In solving our one given system, the first step is to decompose A into the product of an upper triangular matrix U, and a lower triangular matrix L, whose diagonal elements are all ones.

$$A=LU$$

Then

$$A^{-1}=U^{-1}L^{-1}$$

and

$$N(A^{-1}) \leq N(U^{-1})N(L^{-1}).$$

Using the bounds just derived for a triangular matrix, we have

$$N(U^{-1}) \leq \frac{(1+e_1)^{n-1}}{\min|u_{ii}|}, \quad e_1=\max_i \left|\frac{u_{ij}}{u_{ii}}\right| \text{ for all } j$$

and

$$N(L^{-1}) \leq \frac{(1+e_2)^{n-1}}{1}, \quad e_2=\max_{ij} |l_{ij}|.$$

The quantities in the last wo bounds are readily available from the decomposition of A and thus with very little extra work we will have a bound for the error in the solution vector z without needing an approximate inverse. Since for any norm, $N(A^{-1}b-z) \leq N(A^{-1})N(b-Az)$, we have

THEOREM 5.6.

$$\mathrm{N_{rs}(A^{-1}b-z)} \leq \frac{[(1+\mathrm{e_1})(1+\mathrm{e_2})]^{n-1}}{\min|\mathrm{u_{ii}}|}\,\mathrm{N_{rs}(b-Az)}.$$

How much of an overestimate this may be again results primarily from using

$$N(AB) \leq N(A)N(B)$$

as mentioned in section 1. We know that the residual is usually much smaller than the actual error, however, and this fact may help make this bound practical.

Examining in itself the LU decomposition of A will give us a slight refinement in the bound for L^{-1}. In the decomposition process we multiply A on the left by a succession of elementary row operators to reduce it to triangular form:

$$M_{n-1}M_{n-2} \cdots M_1 A = U.$$

Hence, $U^{-1}M_{n-1}M_{n-2} \cdots M_1 A = U^{-1}U = I$ and therefore

$$A^{-1} = U^{-1}M_{n-1}M_{n-2} \cdots M_1$$

so that

$$\begin{aligned} L^{-1} &= M_{n-1}M_{n-2} \cdots M_1 \\ &= M \end{aligned}$$

and

$$L = M_1^{-1}M_2^{-1} \cdots M_{n-1}^{-1}.$$

Now each M_k is a lower triangular matrix with elements in only the kth column and ones along the diagonal:

$$M_k = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & -m_{k+1,k} & 1 & & & \\ & & & -m_{k+2,k} & & \ddots & & \\ & & & \vdots & & & \ddots & \\ & & & -m_{nk} & & & & 1 \end{bmatrix}$$

so M_k^{-1} is actually M_k with the signs of the off-diagonal elements reversed. Furthermore

$$L = \begin{bmatrix} 1 & & & & & \\ m_{21} & 1 & & & & \\ m_{31} & m_{32} & 1 & & & \\ m_{41} & m_{42} & \ddots & \ddots & & \\ \vdots & \vdots & & \ddots & \ddots & \\ m_{n1} & m_{n2} & & & m_{n,n-1} & 1 \end{bmatrix}.$$

Now $N(M_k) \leq 1 + \max_i |m_{ik}| \leq 1 + \max_{ii} |m_{ij}|$ for each k so that

$$N(L^{-1}) = N(M)$$
$$\leq \left(1 + \max_{ij} |m_{ij}|\right)^{n-1}$$

and this is exactly what we have found from our other approach for a triangular matrix. (Note that $l_{ij} = m_{ij}$ and these are stored in the memory of the computer.) This could be a large overestimate since we are repeatedly using the inequality we have indicated as a source of trouble:

$$N(M) \leq N(M_{n-1})N(M_{n-2}M_{n-3} \ldots M_1) \leq \ldots \leq N(M_{n-1})N(M_{n-2}) \ldots N(M_1).$$

The refinement comes from not using the second inequality indicated for $N(M_k)$. Let $p_j = \max_i |m_{ij}|$ for each j. Then

$$N(M_k) = 1 + p_k$$

and

$$N(L^{-1}) = N(M) \leq \prod_{j=1}^{n-1} (1 + p_j).$$

Hence we have

$$N(A^{-1}) \leq N(L^{-1})N(U^{-1})$$
$$\leq \prod_{j=1}^{n-1} (1 + p_j) \frac{(1 + e_2)^{n-1}}{\min_i |u_{ii}|},$$

where $e_2 = \max_i \left|\frac{u_{ij}}{u_{ii}}\right|$ for all j.

Finally,

$$N(A^{-1}b - z) \leq \prod_{j=1}^{n-1} (1 + p_j) \frac{(1 + e_2)^{n-1}}{\min |u_{ii}|} N(b - Az)$$

where the notation is the same as before and N is the maximum row sum norm. We therefore have a bound on the error of the approximate solution of a linear algebraic system without the knowledge of the inverse of the system.

Let us now give a numerical example of this bound and also of those from section 2. We will use the example given in J. H. Wilkinson's book *Rounding Errors in Algebraic Processes*, page 111, and we refer the reader to the discussion given there concerning the size of the bounds for the two different values of the righthand sides.

Problem 1:

$$N(A) = .252E+01 \qquad N(b_1) = .912E+00 \qquad N(A^{-1}b_1) = .225E+05$$
$$N(A^{-1}) = .150E+06 \qquad N(b_1 - Az) = .711E-01 \qquad N(A^{-1}b_1 - z) = .151E+04$$

$$1.\ N(A^{-1}) \leq \prod_{j=1}^{n} (1 + p_j) \frac{(1 + e_2)^{n-1}}{\min |u_{ii}|} = P = .867E+06$$

$$\leq \frac{[(1 + e_1)(1 + e_2)]^{n-1}}{\min |u_{ii}|} = .112E+07$$

$$N(X) = .161E+06$$

2. $N(A^{-1}b_1-z) \leq N(A^{-1})N(b_1-Az) \quad = D = .107E+05$

$\text{or } PN(b_1-Az) \quad = E = .616E+05$

$\text{or } \frac{N(X)N(b_1-Az)}{1-N(Y)} \quad = F = .138E+05$

$\text{or } \frac{N(X)N(Y)N(b_1)}{1-N(Y)} \quad = G = .307E+05$

3. $\frac{N(A^{-1}b_1-z)}{N(A^{-1}b_1)} \leq \frac{D}{N(A^{-1}b_1)} \quad = .475E+00$

$\text{or } \frac{E}{N(A^{-1}b_1)} \quad = .274E+01$

$\text{or } \frac{F}{N(A^{-1}b_1)} \quad = .614E+00$

$\text{or } \frac{G}{N(A^{-1}b_1)} \quad = .137E+01$

$\text{or } N(I-XA) \quad = .125E+00$

$= \text{actually} \quad = .673E-01$

4. $N(A^{-1}b_1-z) \geqslant \frac{N(b_1-Az)}{N(A)} \quad = .283E-01$

Problem 2:

$N(A) = .252E+01 \qquad N(b_2) = .876E+00 \qquad N(A^{-1}b_2) = .915E+00$

$N(A^{-1}) = .150E+06 \qquad N(b_2-Az) = .231E-06 \qquad N(A^{-1}b_2-z) = .607E-02$

(Same as previous problem except that the third element of b_1 was decreased by .264139.)

1. $N(A^{-1}b_2-z) \leq N(A^{-1})N(b_2-Az) \quad = D = .348E-01$

$\text{or } PN(b_2-Az) \quad = E = .200E+00$

$\text{or } \frac{N(X)N(b_2-Az)}{1-N(Y)} \quad = F = .449E-01$

$\text{or } \frac{N(X)N(Y)N(b_2)}{1-N(Y)} \quad = G = .295E+05$

$\geqslant \frac{N(b_2-Az)}{N(A)} \quad = H = .919E-07$

2. $\frac{N(A^{-1}b_2-z)}{N(A^{-1}b_2)} \leq \frac{D}{N(A^{-1}b_2)} \quad = .380E-01$

$\text{or } \frac{E}{N(A^{-1}b_2)} \quad = .219E+00$

$\text{or } \frac{F}{N(A^{-1}b_2)} \quad = .491E-01$

$\text{or } \frac{G}{N(A^{-1}b_2)} \quad = .322E+05$

$\text{or } N(I-XA) \quad = .125E+00$

$= \text{actually} \quad = .668E-02$

6. Evaluation of Computer Programs

In this section, the second part of our research, we would like to use the mathematics we have developed in our first five sections in the evaluation of the performance of some computer programs. We shall set up a problem, choose some programs, and select some test cases to use as the data for each program.

Since calculating the inverse of a matrix usually includes the solution of a linear algebraic system, we have chosen the inversion of a matrix for our problem. The programs we will use were chosen primarily on the basis of availability. Some were on the computer library tapes at the University of Maryland, some were in the literature, and others were received from individuals in the field. It seems that almost all of the programs being used today employ some form of Gauss elimination to do matrix inversion, so we have confined ourselves to this one direct method. The test matrices were chosen to cover a variety of condition numbers, from a small value where all programs should perform quite well, to higher values where some difficulties could be encountered.

Our main concern will be with the accuracy of the result; i.e., with the smallness of $N(A^{-1}-X)$, where X is the approximate inverse produced by a computer program. (This, of course, is not the only criterion of success but only the originator of the problem can decide exactly what is important to him.) Hence, we use the error bound derived in section 1, that

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)}, \qquad Y=I-AX,$$

as our basis for evaluation. We will also limit ourselves to the righthand residual matrix since it is usually smaller, although, as will be seen, this is not always true. It has been frequently pointed out that theoretical error estimates are generally overestimates and sometimes of little practical value. For comparison purposes, however, they serve quite well and furthermore, we will show that this particular error bound is actually quite close to reality. It is true that we have considered only a sampling of computer programs and test problems but we feel they are quite representative and that further work will only support our conclusions.

There are many ways of arriving at a "condition number" for a matrix. The one we shall use is called the P-condition number:

$$P(A)=\left|\frac{\lambda}{\mu}\right|$$

where λ is an eigenvalue of largest modulus and μ, of smallest [14].

Before going into the details of the test matrices and the computer programs, there are some very important items that should be noted first. As we have indicated in many of the previous sections, the possibility of the occurrence of some kind of error must always be kept in mind. Checking the input, execution, and the output of a computer run may be tedious but it is quite essential. We have chosen matrices that can be generated within the machine, used two kinds of checks throughout the execution, and have the exact inverses available to check the final results. The main check on the execution of the program (and the machine) is achieved by including an extra row and column in the original matrix before entering the inversion routine. If the additional column is made up of the negative of the sum of the elements of each of the rows of the matrix, then the row sums of the augmented matrix should always be zero. If we put a 1 in the $(n+1, n+1)$ position and zeros in the other positions of the $n+1$ row, the last column in the computed inverse should be all ones. Let $b=(1, 1, \ldots, 1)^T$. Then

$$\begin{bmatrix} A & -Ab \\ 0 & 1 \end{bmatrix}^{-1}=\begin{bmatrix} A^{-1} & b \\ 0 & 1 \end{bmatrix}.$$

Hence, the maximum difference from unity will give us a good indication if anything has gone wrong. What may have happened (machine failure, large rounding errors, etc.) will vary but at least we have something definite to tell us to beware of the results. Another check we used was simply including an additional column of ones in the second matrix of any product, AB; the result will include the row sums of A, which can easily be checked by hand.

A final check on any run of a computer program should always include a test problem at the beginning so that when we examine the output, we can see if the machine performed properly and if the program is doing what it is supposed to do. It can also serve as a final check on ourselves; for example, if we had dimensioned the variables properly according to all the subroutines we may be using. Another check on the output of a computer run comes from a relation that exists between the norms we mentioned in the Introduction, the Frobenius norm, N_F, and the maximum element norm, N_M. It can be shown that $N_F \leq N_M$. Hence, using this relation can give us more information about the validity of the results. We have found all these checks both necessary and effective.

One final note on avoiding input errors. In ill-conditioned problems, if the data are not exact, the results could be very far from correct. Hence, all our data are integers, exactly represented in the machine. This is accomplished by multiplying each matrix by an appropriate scalar, which will be indicated in the next subsection.

The Test Matrices

1. T_{10}^4

This is the fourth power of the 10×10 tridiagonal matrix with -2 on the diagonal, 1 above and below the diagonal, and 0 elsewhere.

$$T = (t_{ij}),\ t_{ij} = \begin{cases} -2, & i = j \\ 1, & |i-j| = 1 \\ 0, & |i-j| \geq 2. \end{cases}$$

$$P(T_{10}^4) \approx (4/\pi^2)^4 \cdot (n^2)^4$$

$$\approx .27E + 07.$$

The elements are all integers.

$$T^{-1} = (b_{ij}),\ b_{ij} = \begin{cases} \dfrac{-i(n-j+1)}{n+1}, & i \leq j \\ b_{ji}, & i > j. \end{cases}$$

2. T_{20}^3

T is the same as above and we just change the dimension to 20 and the power to 3.

$$P(T_{20}^3) \approx (4/\pi^2)^3 \cdot (n^2)^3$$

$$\approx .43E + 07.$$

The elements are all integers.

3. T_{20}^4

T is the same as above and we just change the power to 4.

$$P(T_{20}^4) \approx (4/\pi^2)^4 \cdot (n^2)^4$$

$$\approx .69E + 09.$$

The elements are all integers.

4. A_{100}

This is a 10×10 matrix where $A_k = (1/k)I + J$ and J is the 10×10 matrix of all ones. $P(A_k) = 1 + 10k$. The integer form for use as a test matrix is obviously achieved upon multiplication by k.

$$(kA_k)^{-1} = I - \frac{k}{1 + 10k} J.$$

$$P(A_{100}) = 1001$$

$$\approx .10E + 04.$$

5. A_{1000}

The same as above except that we change the value of k.

$$P(A_{1000}) = 10001$$

$$\approx .10E + 05.$$

6. A_{10000}

The same as above except that we change the value of k.

$$P(A_{10000}) = 100001$$

$$\approx .10E + 06.$$

7. H_6

H_n is the $n \times n$ Hilbert matrix.

$$H_n = (h_{ij}),\ h_{ij} = (i + j - 1)^{-1}$$

and

$$\log_e [P(H_n)] \approx 3.5n.$$

The integer form for use as a test matrix is achieved upon multiplication by the least common multiple of $(1, 2, 3, \ldots, 2n - 1)$.

$$H_n^{-1} = (b_{ij})$$

where

$$b_{ij} = \frac{(-1)^{i+j}(n + i - 1)!(n + j - 1)!}{(i + j - 1)[(i - 1)!(j - 1)!]^2 (n - i)!(n - j)!}.$$

$$P(H_6) \approx .13E + 10$$

and the multiplier for integer input is 27720.

8. H_8

This is the 8×8 Hilbert matrix.

$$P(H_8) \approx .14E + 13$$

and the scalar multiplier for integer input is 360360.

9. H_{10}

This is the 10×10 Hilbert matrix.

$$P(H_{10}) \approx .16E+16$$

and the scalar multiplier for integer input is 232792560.

The Computer Programs

1. LEQ

A FORTRAN subroutine used to solve the matrix equation $AX=B$ and to evaluate the determinant of A. It was written by Max Goldstein of the AEC Computing and Applied Mathematics Center at the Courant Institute of Mathematical Sciences, New York University. The Gauss elimination method is used. The matrices are normalized row-wise by dividing by the largest element of $A(I, J)$ in that row, then the A matrix is reduced to triangular form by $(N-1)$ transformations using a pivotal condensation process after which $X(I, K)$ is computed by a back-substitution process. This transforms B into X and leaves the product of the diagonal elements as the determinant of A.

2. MATH PACK, GJR

A FORTRAN subroutine, one of the Univac 1108 Math Pack programs, available on the library tapes at the University of Maryland computing center. It solves simultaneous equations, computes a determinant, or inverts a matrix or any combination of the three above by using a Gauss-Jordan elimination technique with column pivoting.

3. MATH PACK, MXHOI

A FORTRAN subroutine, one of the Univac 1108 Math Pack programs, available on the library tapes at the University of Maryland computing center. It improves the accuracy of the inverse of a matrix by an iterative method which has cubic convergence.

4. MIDAS

A FORTRAN and ALGOL package to solve general nonsingular systems of linear algebraic equations, invert matrices, and compute determinants. Error bounds on the solution or inverse are available as an option. It was written by Peter A. Businger of Bell Telephone Laboratories, Inc., Murray Hill, New Jersey. The error bound is a bound on the distance between any element of the true inverse and the corresponding element of the computed inverse (unless the bound equals -1, in which case no bound is available). Gaussian elimination with partial pivoting is used to decompose the $N \times N$ input matrix into the product of a lower and an upper triangular matrix (LU decomposition). The magnitude of intermediate results is estimated; in case of alarming growth the program switches to complete pivoting. When solving a system of equations $Ax=b$, the accuracy of the solution obtained from the triangular systems is improved by iteration; in the case of matrix inversion the iteration is omitted for the sake of computational efficiency. (We note that the error bound is essentially $N(X)N(Y)/[1-N(Y)]$, $Y=I-AX$.)

5. MINV

A FORTRAN subroutine, one of the IBM System/360 Scientific Subroutine Package. It inverts a general matrix by the standard Gauss-Jordan method. The determinant is also calculated. A determinant of zero indicates a singular matrix.

6. OMNITAB, INVERT

OMNITAB is a general-purpose computer program for statistical and numerical analysis developed at the National Bureau of Standards by Joseph Hilsenrath et al. Now available in an ASA FORTRAN version, OMNITAB allows the user to communicate with a computer in an efficient manner by means of simple English sentences. INVERT is the calling word for the matrix inversion routine used and an error bound is given: $N(X)N(Y)/[1-N(Y)]$.

406-680 O - 71 - 5

7. SOLVE

A FORTRAN program by Cleve Moler given in the book *Computer Solution of Linear Algebraic Systems* by George Forsythe and Cleve Moler. It uses Gauss elimination with partial pivoting and has a subroutine IMPRUV, which can be called to improve the solution of a linear algebraic system. Appropriate messages for various kinds of singularity are available. It is presently undergoing some changes to increase efficiency in most FORTRAN systems although these changes should not materially alter the numerical behavior.

8. LSD

This is a program, with automatic error bounding, for solving a system of linear equations. This is equivalent to finding the solution, with error bounds, of the matrix equation $Ax=b$ where A is an $N\times N$ matrix and b and x are $N\times 1$ vectors. The error bounding is done with interval arithmetic (I.A.). The program permits interval input and hence errors may be included which arise from uncertainty in the data. A second entry to LSD is provided for solving several sets of equations with the same A. Thus the program is efficient in finding the inverse of A, with error bounds, by using successively for b the columns of the identity matrix. Output from the program includes an upper and lower bound on the solution vector. Therefore, the number of significant figures in the answer may be simply read from the output. LSD uses two FORTRAN subroutines, a FAP function, and a FAP subroutine in addition to interval arithmetic subroutines coded in FAP. The program requires only standard FORTRAN II software. The double precision arithmetic hardware feature of the IBM 7094 is used, thus restricting the use of LSD, in its present form to this machine. It was written by Eldon Hansen and Roberta Smith.

We intended to include four other programs in our evaluation:

1. STAT PACK, JIM

A FORTRAN subroutine, one of the Univac 1108 STAT-PACK programs, available on the library tape at the University of Maryland computing center. Apparently, this subroutine never functioned properly and the subroutine actually calls GJR of MATH PACK.

2. SPVMTX

A single precision FORTRAN IV program for inverting a matrix or solving a set of linear equations. To a program from the SHARE library (7090–F1 3180INV1 Single Precision Matrix Inversion with Selective Pivoting, written by A. R. Sadaka), Sally T. Peavy, National Bureau of Standards, incorporated accuracy checks. We belatedly discovered that this is the routine used by INVERT of OMNITAB. This was confirmed by the identical outputs of both programs.

3. LEQU

A FORTRAN subroutine to approximate the solution X of the system $AX=B$ of linear algebraic equations, where A is an $N\times N$ matrix and B and X are $N\times M$ matrices. This program is faster but less accurate than subroutine LEQUN, which should normally be preferred. It was written by William Kahan, who was then at the Institute of Computer Science, University of Toronto (presently at the University of California, Berkeley). We found that this program depends heavily on other routines in the University of Toronto 7094 library and we were unable to delete this dependence or acquire the other routines at this time.

4. ORTHO1

An ALGOL subroutine (with iterative improvement) to invert a matrix. It was written by F. L. Bauer and essentially uses Gauss elimination but with a suitably weighted combination of rows

used for elimination instead of a single row. We were unable, at this time, to run the program in ALGOL or to translate it into FORTRAN.

We would like to mention one other program that exists, one which solves equations exactly. It is called SOLVER and was written in FORTRAN by Dr. Morris Newman of the National Bureau of Standards. It employs a congruential method and calculates the exact solution of $Ax = b$ where the elements of A and b must be integers. It will solve systems in which A is a square matrix at most 100×100 and the elements of A and b are numerically less than 10^{20}. This method is not at all sensitive to the condition of A, but it can be time-consuming for large systems. It has successfully inverted the Hilbert matrices of all orders up to and including $n = 13$ (a limit imposed by considerations of time). Since our test matrices all have known inverses, it was not necessary to run this program.

The Results

In the following tables we list some of the information that we had available from the execution of each program for the test problems. We rank the programs according to the smallness of the error bound. In some cases, the values of this bound were quite close so the difference in the rank may sometimes be slightly artificial. We did not include MXHOI in the rankings and this will be explained later. For our estimates we used the Frobenius norm and the maximum element norm and we group our results accordingly. In the cases where $N(I-AX)$ was greater than 1, our error bounds do not apply and so we leave that row blank. With regard to the check vector carried throughout the inversion process, we subtracted this from the vector $b = (1, 1, \ldots, 1)^T$ and list the maximum absolute value of this difference. We note that for a check matrix (T_6 with a P-condition number of approximately 15) all the programs performed extremely well, the only difference, if any, being in the last digit. We will discuss the interval arithmetic program of Hansen and Smith separately, immediately after presenting the results of the other programs.

All the calculations, except those for the interval arithmetic program, were performed on the Univac 1108 at the University of Maryland between March 10 and March 22, 1970. In use at this time was version 23.65.74:N of the EXEC 8 system.

TABLE 1. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.21 $E-02$	.13 $E-01$	.29 $E+02$	.30 $E+03$	.12 $E-02$	.23 $E+01$	2
MATH PAK	GJR	.16 $E-01$	.69 $E+00$	.12 $E+04$	.51 $E+05$	.86 $E-01$	.34 $E-01$	6
	MXHOI	.21 $E-02$	.25 $E-01$	.66 $E+02$	.60 $E+03$	.29 $E-02$	.35 $E+04$	
MIDAS		.83 $E-02$	.29 $E-01$	.15 $E+03$	.68 $E+03$	.65 $E-02$	.12 $E+00$	4
MINV		.87 $E-02$	.29 $E-01$	.15 $E+03$	.69 $E+03$	.66 $E-02$	.27 $E-01$	4
OMNITAB (INVERT)		.12 $E-01$	.12 $E+01$				.46 $E-01$	7
SOLVE	no IMPRUV	.35 $E-02$	.86 $E-02$	.76 $E+02$	.20 $E+03$	.33 $E-02$	.73 $E+00$	3
	IMPRUV	.75 $E-08$	.81 $E-02$	.15 $E-03$	.19 $E+03$	.65 $E-08$	.94 $E-02$	1

Matrix T_{10}^4

Condition number .27 $E+07$

Norm Frobenius

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

*Difference from 1.

TABLE 2. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.20 $E-02$	.13 $E-01$	.22 $E+03$	.12 $E+04$	.25 $E-02$	.93 $E+01$	3
MATH PAK	GJR	.12 $E-02$	.66 $E+01$				.20 $E-01$	6
	MXHOI	.41 $E-02$	.31 $E-01$	.31 $E+03$	.28 $E+04$	.35 $E-02$	.65 $E+04$	
MIDAS		.38 $E-02$	.20 $E-01$	.24 $E+03$	.18 $E+04$	.27 $E-02$	.57 $E+01$	4
MINV		.45 $E-02$	.31 $E-01$	.36 $E+03$	.29 $E+04$	.41 $E-02$	.23 $E-01$	5
OMNITAB (INVERT)		.17 $E-02$	.70 $E+01$				.36 $E-01$	6
SOLVE	no IMPRUV	.17 $E-02$	.15 $E-01$	.18 $E+03$	.14 $E+04$	.21 $E-02$	.58 $E+01$	2
	IMPRUV	.75 $E-08$	.99 $E-02$	.60 $E-03$	.90 $E+03$	.68 $E-08$	.10 $E-01$	1

Matrix T^3_{20}
Condition number .43 $E+07$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leqslant \frac{N(XY)}{1-N(Y)} \leqslant \frac{N(X)N(Y)}{1-N(Y)}$$

$Y = I-AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 3. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.47 $E+01$	.78 $E+01$				.22 $E+05$	
MATH PAK	GJR	.71 $E+00$	.36 $E+04$				.31 $E+01$	
	MXHOI		overflow					
MIDAS		.51 $E+00$	.28 $E+01$				.15 $E+04$	
MINV		.69 $E+00$	.32 $E+01$				.21 $E+01$	
OMNITAB (INVERT)		.37 $E+00$	.98 $E+04$				.45 $E+01$	
SOLVE	no IMPRUV	.71 $E+00$	.10 $E+01$				.30 $E+04$	
	IMPRUV	.13 $E-02$	.18 $E+01$				.14 $E+02$	

Matrix T^4_{20}
Condition number .69 $E+09$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leqslant \frac{N(XY)}{1-N(Y)} \leqslant \frac{N(X)N(Y)}{1-N(Y)}$$

$Y = I-AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leqslant \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 4. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.75 $E-06$	.17 $E-04$	.12 $E-05$	.50 $E-04$	.39 $E-06$	.20 $E-04$	7
MATH PAK	GJR	.26 $E-05$	.42 $E-04$	.84 $E-06$	.13 $E-03$	.28 $E-06$	.41 $E-04$	6
	MXHOI	.26 $E-05$	.42 $E-04$	.84 $E-06$	.13 $E-03$	.28 $E-06$	.41 $E-04$	
MIDAS		.86 $E-05$	.66 $E-04$	.13 $E-06$	.20 $E-03$	.43 $E-07$	.20 $E-04$	3
MINV		.30 $E-07$	.36 $E-04$	.39 $E-07$	.11 $E-03$	.13 $E-07$	.19 $E-04$	2
OMNITAB (INVERT)		.25 $E-05$	.41 $E-04$	.83 $E-06$	.12 $E-03$	.28 $E-06$	.42 $E-04$	5
SOLVE	no IMPRUV	.26 $E-05$	.15 $E-04$	.82 $E-06$	.44 $E-04$	.27 $E-06$	.18 $E-04$	4
	IMPRUV	.75 $E-08$	.72 $E-05$	.14 $E-07$	.22 $E-04$	.47 $E-08$	.72 $E-05$	1

Matrix A_{100}
Condition number .10 $E+04$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$Y = I - AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 5. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.34 $E-04$	.18 $E-03$	.13 $E-04$	.54 $E-03$	.44 $E-05$	.18 $E-03$	7
MATH PAK	GJR	.12 $E-03$	.33 $E-03$	.65 $E-05$	.10 $E-02$	.22 $E-05$	.34 $E-03$	4
	MXHOI	.12 $E-03$	.33 $E-03$	.65 $E-05$	.10 $E-02$	.22 $E-05$	.34 $E-03$	
MIDAS		.67 $E-04$	.30 $E-03$	.89 $E-07$	.91 $E-03$	.30 $E-07$	.24 $E-03$	3
MINV		.30 $E-07$	.29 $E-03$	.36 $E-07$	.86 $E-03$	.12 $E-07$	.18 $E-03$	2
OMNITAB (INVERT)		.14 $E-04$	.43 $E-03$	.65 $E-05$	.13 $E-02$	.22 $E-05$	.43 $E-03$	4
SOLVE	no IMPRUV	.14 $E-04$	.15 $E-03$	.65 $E-05$	.46 $E-03$	.22 $E-05$	.16 $E-03$	4
	IMPRUV	.15 $E-07$	.84 $E-04$	.94 $E-08$	.25 $E-03$	.31 $E-08$	.84 $E-04$	1

Matrix A_{1000}
Condition number .10 $E+05$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$Y = I - AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 6. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.42 $E-03$	.13 $E-02$	.30 $E-03$	.39 $E-02$	.10 $E-03$	.18 $E-02$	7
MATH PAK	GJR	.22 $E-06$	.29 $E-02$	.90 $E-04$	.88 $E-02$	.30 $E-04$	.30 $E-02$	4
	MXHOI	.80 $E-06$	.54 $E-02$	.15 $E-03$	.16 $E-01$	.51 $E-04$	.31 $E+01$	
MIDAS		.11 $E-02$	.27 $E-02$	.29 $E-06$	.81 $E-02$	.96 $E-07$	.17 $E-02$	3
MINV		.30 $E-07$	.32 $E-02$	.36 $E-07$	.95 $E-02$	.12 $E-07$	.16 $E-02$	2
OMNITAB (INVERT)		.60 $E-07$	.41 $E-02$	.90 $E-04$	.12 $E-01$	.30 $E-04$	.41 $E-02$	4
SOLVE	no IMPRUV	.30 $E-07$	.16 $E-02$	.90 $E-04$	.48 $E-02$	.30 $E-04$	.20 $E-02$	4
	IMPRUV	.15 $E-07$	.48 $E-03$	.10 $E-07$	.14 $E-02$	.35 $E-08$	.48 $E-03$	1

Matrix A_{10000}
Condition number .10 $E+06$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$Y=I-AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 7. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.20 $E-01$	.19 $E-01$	.92 $E+00$	.63 $E+01$	.28 $E-02$	.47 $E-01$	2
MATH PAK	GJR	.30 $E-02$	.30 $E-01$	.22 $E+01$	.10 $E+02$	.67 $E-02$	.31 $E-01$	7
	MXHOI	.11 $E-01$	.34 $E-01$	.12 $E+01$	.12 $E+02$	.36 $E-02$	.56 $E+03$	
MIDAS		.10 $E-01$	.28 $E+00$	.18 $E+01$	.13 $E+03$	.67 $E-02$	.41 $E-01$	6
MINV		.86 $E-02$	.56 $E-01$	.11 $E+01$	.20 $E+02$	.36 $E-02$	.56 $E-01$	3
OMNITAB (INVERT)		.10 $E-01$	.40 $E-01$	.17 $E+01$	.14 $E+02$	.52 $E-02$	.36 $E-01$	5
SOLVE	no IMPRUV	.16 $E-01$	.18 $E-01$	.16 $E+01$	.61 $E+01$	.50 $E-02$	.65 $E-01$	4
	IMPRUV	.00	.96 $E-02$	.29 $E-05$	.32 $E+01$	.88 $E-08$	.96 $E-02$	1

Matrix H_6
Condition number .13 $E+10$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$Y=I-AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 8. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ			overflow					
MATH PAK	GJR	.93 $E+00$	.20 $E+02$				.89 $E+01$	
	MXHOI		overflow					
MIDAS		.77 $E+01$	.16 $E+03$				.10 $E+03$	
MINV		.22 $E+02$	.26 $E+02$				.23 $E+02$	
OMNITAB (INVERT)		.67 $E+00$	.66 $E+01$				.52 $E+01$	
SOLVE	no IMPRUV	.19 $E+01$	.20 $E+01$				.26 $E+02$	
	IMPRUV	.16 $E+02$	.10 $E+03$				.35 $E+03$	

Matrix H_8
Condition number .14 $E+13$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 9. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ			overflow					
MATH PAK	GJR	.79 $E+01$	.14 $E+04$				.67 $E+02$	
	MXHOI		overflow					
MIDAS		.89 $E+01$	.35 $E+02$				.96 $E+01$	
MINV		.18 $E+02$	.24 $E+02$				.22 $E+04$	
OMNITAB (INVERT)		.17 $E+02$	.76 $E+02$				.15 $E+02$	
SOLVE	no IMPRUV	.27 $E+02$	.39 $E+02$				.16 $E+04$	
	IMPRUV	.47 $E+03$	.59 $E+03$				.85 $E+05$	

Matrix H_{10}
Condition number .16 $E+16$
Norm Frobenius

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 10. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.21 $E-02$	.44 $E-01$	.53 $E+02$	.19 $E+04$	.13 $E-02$	.66 $E+01$	5
MATH PAK	GJR	.16 $E-01$	.18 $E+01$				.11 $E+00$	6
	MXHOI	.21 $E-02$	.91 $E-01$	.22 $E+03$	.42 $E+04$	.58 $E-02$	.86 $E+04$	
MIDAS		.83 $E-02$	.13 $E+00$	.29 $E+03$	.59 $E+04$	.79 $E-02$	.47 $E+00$	4
MINV		.87 $E-02$	.70 $E-01$	.27 $E+03$	.31 $E+04$	.71 $E-02$	.92 $E-01$	3
OMNITAB (INVERT)		.12 $E-01$	.26 $E+01$				.12 $E+00$	6
SOLVE	no IMPRUV	.35 $E-02$	.35 $E-01$	.14 $E+03$	.15 $E+04$	.35 $E-02$	.23 $E+01$	2
	IMPRUV	.75 $E-08$	.27 $E-01$	.46 $E-03$	.11 $E+04$	.11 $E-07$	.29 $E-01$	1

Matrix T_{10}^{4}

Condition number .27 $E+07$

Norm Maximum element

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 11. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.20 $E-02$	.68 $E-01$	.45 $E+03$	.12 $E+05$	.28 $E-02$	.30 $E+02$	3
MATH PAK	GJR	.12 $E-02$	.23 $E+02$				.76 $E-01$	6
	MXHOI	.41 $E-02$	.13 $E+00$	.96 $E+03$	.25 $E+05$	.63 $E-02$	.19 $E+05$	
MIDAS		.38 $E-02$	.77 $E-01$	.48 $E+03$	.14 $E+05$	.30 $E-02$	.28 $E+02$	4
MINV		.45 $E-02$	.12 $E+00$	.75 $E+03$	.24 $E+05$	.50 $E-02$	.11 $E+00$	5
OMNITAB (INVERT)		.17 $E-02$	.17 $E+02$				.12 $E+00$	6
SOLVE	no IMPRUV	.17 $E-02$	.68 $E-01$	.37 $E+03$	.12 $E+05$	.23 $E-02$	.19 $E+02$	2
	IMPRUV	.75 $E-08$	.42 $E-01$	.22 $E-02$	.74 $E+04$	.13 $E-07$	.42 $E-01$	1

Matrix T_{20}^{3}

Condition number .43 $E+07$

Norm Maximum element

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 12. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEG		.47 $E+01$	.31 $E+02$				.81 $E+05$	
MATH PAK	GJR	.71 $E+00$	.11 $E+05$				.11 $E+02$	
	MXHOI		overflow					
MIDAS		.51 $E+00$	.12 $E+02$				.73 $E+04$	
MINV		.69 $E+00$	.83 $E+01$				.79 $E+01$	
OMNITAB (INVERT)		.37 $E+00$	.30 $E+05$				.16 $E+02$	
SOLVE	no IMPRUV	.71 $E+00$	.38 $E+01$				.15 $E+05$	
	IMPRUV	.13 $E-02$	.73 $E+01$				.44 $E+02$	

Matrix T^3_{20}

Condition number .69 $E+09$

Norm Maximum element

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$Y = I-AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 13. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.73 $E-06$	.29 $E-04$	.10 $E-04$	.26 $E-03$	.11 $E-05$	.38 $E-04$	7
MATH PAK	GJR	.26 $E-05$	.69 $E-04$	.76 $E-05$	.62 $E-03$	.84 $E-06$	.66 $E-04$	6
	MXHOI	.26 $E-05$	.69 $E-04$	.76 $E-05$	.62 $E-03$	.84 $E-06$	.66 $E-04$	
MIDAS		.86 $E-05$	.17 $E-03$	.41 $E-06$	.16 $E-02$	.46 $E-07$	.33 $E-04$	3
MINV		.30 $E-07$	.11 $E-03$	.12 $E-06$	.98 $E-03$	.13 $E-07$	.20 $E-04$	2
OMNITAB (INVERT)		.25 $E-05$	.57 $E-04$	.74 $E-05$	.51 $E-03$	.82 $E-06$	.58 $E-04$	5
SOLVE	no IMPRUV	.26 $E-05$	.31 $E-04$	.73 $E-05$	.28 $E-03$	.82 $E-06$	.25 $E-04$	4
	IMPRUV	.75 $E-08$	.76 $E-05$	.42 $E-07$	.69 $E-04$	.47 $E-08$	.76 $E-05$	1

Matrix A_{100}

Condition number .10 $E+04$

Norm Maximum element

*Difference from 1.

X = Approximate inverse

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$Y = I-AX$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 14. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.34 $E-04$	.34 $E-03$	.58 $E-04$	.31 $E-02$	.64 $E-05$	.35 $E-03$	7
MATH PAK	GJR	.12 $E-03$	.57 $E-03$	.59 $E-04$	.52 $E-02$	.65 $E-05$	.68 $E-03$	4
	MXHOI	.12 $E-03$	.57 $E-03$	.59 $E-04$	.52 $E-02$	.65 $E-05$	.68 $E-03$	
MIDAS		.67 $E-04$	.80 $E-03$	.54 $E-06$	.72 $E-02$	.60 $E-07$	.37 $E-03$	3
MINV		.30 $E-07$	.85 $E-03$	.93 $E-07$	.77 $E-02$	.10 $E-07$	.18 $E-03$	2
OMNITAB (INVERT)		.14 $E-04$	.63 $E-03$	.58 $E-04$	.57 $E-02$	.65 $E-05$	.59 $E-03$	4
SOLVE	no IMPRUV	.14 $E-04$	.29 $E-03$	.58 $E-04$	.26 $E-02$	.65 $E-05$	.27 $E-03$	4
	IMPRUV	.15 $E-07$	.88 $E-04$	.18 $E-07$	.79 $E-03$	.20 $E-08$	.88 $E-04$	1

Matrix A_{1000}
Condition number .10 $E+05$
Norm Maximum element

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 15. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.42 $E-03$	.34 $E-02$	.21 $E-02$	.31 $E-01$	.24 $E-03$	.30 $E-02$	7
MATH PAK	GJR	.22 $E-06$	.59 $E-02$	.82 $E-03$	.53 $E-01$	.91 $E-04$	.65 $E-02$	5
	MXHOI	.80 $E-06$	.96 $E-02$	.70 $E-03$	.87 $E-01$	.79 $E-04$	.63 $E+01$	
MIDAS		.11 $E-02$	.60 $E-02$	.12 $E-05$	.54 $E-01$	.13 $E-06$	.25 $E-02$	3
MINV		.30 $E-07$	.98 $E-02$	.11 $E-06$	.89 $E-01$	.12 $E-07$	.16 $E-02$	2
OMNITAB (INVERT)		.60 $E-07$	.61 $E-02$	.82 $E-03$	.55 $E-01$	.91 $E-04$	.62 $E-02$	5
SOLVE	no IMPRUV	.30 $E-07$	.29 $E-02$	.81 $E-03$	.26 $E-01$	.91 $E-04$	.40 $E-02$	4
	IMPRUV	.15 $E-07$	.49 $E-03$	.31 $E-07$	.45 $E-02$	.35 $E-08$	.49 $E-03$	1

Matrix A_{10000}
Condition number .10 $E+06$
Norm Maximum element

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

TABLE 16. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ		.20 $E-01$	.57 $E-01$	.26 $E+01$	.58 $E+02$	.29 $E-02$	.12 $E+00$	2
MATH PAK	GJR	.30 $E-02$	.12 $E+00$	.68 $E+01$	.14 $E+03$	.80 $E-02$	.97 $E-01$	6
	MXHOI	.11 $E-01$	.11 $E+00$	.32 $E+01$	.12 $E+03$	.37 $E-02$	.12 $E+04$	
MIDAS		.10 $E-01$	.86 $E+00$	.27 $E+02$	.61 $E+04$	.52 $E-01$	.16 $E+00$	7
MINV		.86 $E-02$	.19 $E+00$	.37 $E+01$	.23 $E+03$	.46 $E-02$	.22 $E+00$	3
OMNITAB (INVERT)		.10 $E-01$	.91 $E-01$	.51 $E+01$	.96 $E+02$	.58 $E-02$	.12 $E+00$	5
SOLVE	no IMPRUV	.16 $E-01$	.61 $E-01$	.49 $E+01$	.62 $E+02$	.54 $E-02$	.13 $E+00$	4
	IMPRUV	.00	.23 $E-01$	.11 $E-04$	.22 $E+02$	.11 $E-07$	.23 $E-01$	1

Matrix H_6
Condition number .13 $E+10$
Norm Maximum element

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

*Difference from 1.

TABLE 17. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ			overflow					
MATH PAK	GJR	.93 $E+00$	.67 $E+02$				.25 $E+02$	
	MXHOI		overflow					
MIDAS		.77 $E+01$	.49 $E+03$				.67 $E+03$	
MINV		.22 $E+02$	.81 $E+02$				.80 $E+02$	
OMNITAB (INVERT)		.67 $E+00$	.29 $E+02$				.16 $E+02$	
SOLVE	no IMPRUV	.19 $E+01$	.80 $E+01$				.66 $E+02$	
	IMPRUV	.16 $E+02$	.26 $E+03$				.12 $E+04$	

Matrix H_8
Condition number .14 $E+13$
Norm Maximum element

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

*Difference from 1.

TABLE 18. *Summary of results*

Program		Max ELT* of check vector	$N(I-AX)$	$\frac{N(XY)}{1-N(Y)}$	$\frac{N(X)N(Y)}{1-N(Y)}$	Relative error	$N(I-XA)$	Rank
LEQ			overflow					
MATH PAK	GJR	.79 $E+01$	.58 $E+04$				.26 $E+03$	
	MXHOI		overflow					
MIDAS		.89 $E+01$	.12 $E+03$				.61 $E+02$	
MINV		.18 $E+02$	.12 $E+03$				.94 $E+02$	
OMNITAB (INVERT)		.17 $E+02$	.20 $E+03$				.79 $E+02$	
SOLVE	no IMPRUV	.27 $E+02$	.13 $E+03$				.40 $E+04$	
	IMPRUV	.47 $E+03$	.24 $E+04$				.25 $E+06$	

Matrix H_{10}
Condition number .16 $E+16$
Norm Maximum element

*Difference from 1.

X = Approximate inverse

$Y = I - AX$

$$N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$$

$$\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$$

For the program LSD, using interval arithmetic, we found excellent results for all the matrices except H_8, H_{10}, and T^4_{20}. By excellent results we mean the upper and lower values given for each element of the inverse agreed to 7 or 8 digits with each other and with the exact value. For H_8, the spread between the upper and lower values was quite different. A typical situation is given in the first row below, and the worst spread in the second row.

A^{-1} Lower	Exact	A^{-1} Upper
.12533643 $E-03$	.17760000 $E-03$	.22986282 $E-03$
$-$.16101655 $E+01$	$-$.80000000 $E+00$	$+$.10338515 $E-01$

In view of the fact that none of the other programs solved this problem, these results are not bad.

For the 10×10 Hilbert matrix, the following error message was printed out: SINGULAR MATRIX INCLUDED IN INPUT MATRIX. (We note that SOLVER produced the exact inverse in both of these cases.)

For T^4_{20} we got results but the spread was so large that they are meaningless. One of the better values was from .26055424 $E+05$ to .56997728 $E+05$ but a more common spread would be similar to the following:

$-$.14807597 $E+07$ to $+$.18754179 $E+07$

or $-$.60322834 $E+08$ to $+$.60828502 $E+08$.

These certainly are not very satisfactory. Let us now turn our attention to the other programs.

Evaluation and Discussion

It is obvious that SOLVE with iterative improvement performed far better than any of the others. The excellent results that are achieved by using this added feature certainly seem to warrant the extra work involved. In each case where it did not succeed in inverting the matrix (H_8, H_{10}, T^4_{20}), the following error message was printed out: ITERATION DID NOT CONVERGE. MATRIX IS NEARLY

SINGULAR. This is most important because it tells us our answers may not be correct. LEQ had floating point overflow, MIDAS had the error bound of -1, and OMNITAB had a negative bound, each of which also indicates doubtful results. MINV and GJR gave no indication that the results might not be correct—a very serious defect. GJR does, in fact, have an error return but it is only for overflow and this is not sufficient. The zero determinant criterion of MINV is also not adequate. Knowing that the condition number is as high as it is for these two matrices, we expected trouble. But for T_{20}^{4} whose condition number is fairly large but whose largest element is 70, we did not anticipate serious difficulty and the message of singularity was needed. In this case, however, LEQ did not have any floating point overflow message.

If we did not know ahead of time that the matrix was poorly conditioned, we could use the iterative improvement scheme if for no other reason than to inform us of doubtful results. In solving a single system of equations where we may not calculate an approximate inverse (from which a residual matrix whose norm being greater than 1 can indicate trouble in spite of the program output), the improvement scheme seems an excellent way of indicating possible poor results.

In the cases where the inversions were successfully accomplished (that is, at least $N(I-AX)$ was less than 1), it seems that MINV, SOLVE (no IMPRUV), and MIDAS were more successful than the others, excluding IMPRUV of course. It is interesting to note that the only program (besides IMPRUV) which had the norm of *both* residuals less than one was MINV. This is certainly a desirable situation. It also points out the large discrepancy that can exist between the two residual matrices and the fact that the righthand residual, $I-AX$, is not always smaller. Hence, a bound on the error matrix can sometimes be available by using the other residual. None of the programs seem to take this possibility into consideration. If we have gone to the trouble of calculating $I-AX$ and get a negative bound, it might be worthwhile to calculate $I-XA$ and see if a bound can be found. This will also indicate the kind of inverse we have. It is still, of course, up to the originator of the problem to decide if this is necessary.

The vagaries of numerical calculations are also clearly indicated by these results. We have cases in the same problem where $N(I-AX) > N(I-XA)$ for one program and the opposite is true for another program; for the same program, we may have $N(I-AX) > N(I-XA)$ for one problem and the opposite for another problem. Hence, one must always be extremely careful in doing numerical work.

The poor and erratic performance of MXHOI led to a brief investigation of this improvement procedure. It never really improved the results and in some cases (A_{10000}) made them worse. We found that no double precision was used at all and, as we indicated in section 3, for a scheme of cubic convergence, multiprecision may be needed, let along double precision. The performance of MXHOI certainly bears this out! The results are completely unreliable and this is why we did not even consider this routine as far as ranking the various programs was concerned.

Finally, we would like to discuss briefly the merits of the bound $N(XY)/[1-N(Y)]$. We have observed in section 1 that using this bound would probably give better results for the relative error than the simpler expression $N(Y)$, where Y is either residual matrix. We note that this is in fact true in every one of our problems where the inversion was successful.

Another and much more important fact is the following. We have attempted in our first part to derive a practical bound for $N(A^{-1}-X)$; i.e., one that is realistic and not misleading. It is clear that using $N(XY)$ is better than $N(X)N(Y)$—there is almost always at least a factor of 10 involved in the improvement. This is certainly worthwhile. But just how close to the actual error is our bound? This question can be partially answered by looking at the lower bound, $N(E)/2N(A)$. This was not included in the tables because the only time it was close to the upper bound was for SOLVE with IMPRUV, which certainly indicates both the effectiveness of this scheme and the reality of the upper bound. A more complete answer comes from calculating the actual error. We chose H_6 and T_{20}^{3} as being of the more difficult type and list the results in the following tables. The comparisons are extremely satisfying.

TABLE 19. *Norms of actual difference*

Program		Maximum Element Norm			Frobenius Norm		
		$\frac{N(XY)}{1-N(Y)}$	$N(A^{-1}-X)$ (actual)	$\frac{N(X)N(Y)}{1-N(Y)}$	$\frac{N(XY)}{1-N(Y)}$	$N(A^{-1}-X)$ (actual)	$\frac{N(X)N(Y)}{1-N(Y)}$
LEQ		.26 E+01	.25 E+01	.58 E+02	.92 E+00	.91 E+00	.63 E+01
MATH PAK	GJR	.68 E+01	.59 E+01	.14 E+03	.22 E+01	.21 E+01	.10 E+02
	MXHOI						
MIDAS		.27 E+02	.37 E+01	.61 E+04	.18 E+01	.13 E+01	.13 E+03
MINV		.37 E+01	.30 E+01	.23 E+03	.11 E+01	.11 E+01	.36 E+02
OMNITAB (INVERT)		.51 E+01	.46 E+01	.96 E+02	.17 E+01	.16 E+01	.14 E+02
SOLVF	no IMPRUV	.49 E+01	.46 E+01	.62 E+02	.16 E+01	.16 E+01	.61 E+01
	IMPRUV	.11 E−04	.60 E−06	.22 E+02	.29 E−05	.14 E−06	.32 E+01

Matrix H_6 X=Approximate inverse $N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$

Condition number .13 E+10 $Y=I-AX$ $\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$

TABLE 20. *Norms of actual difference*

Program		Maximum Element Norm			Frobenius Norm		
		$\frac{N(XY)}{1-N(Y)}$	$N(A^{-1}-X)$ (actual)	$\frac{N(X)N(Y)}{1-N(Y)}$	$\frac{N(XY)}{1-N(Y)}$	$N(A^{-1}-X)$ (actual)	$\frac{N(X)N(Y)}{1-N(Y)}$
LEQ		.45 E+03	.42 E+03	.12 E+05	.22 E+03	.22 E+03	.12 E+04
MATH PAK	GJR**	.40 E+03	.37 E+03	.14 E+05	.19 E+03	.19 E+03	.18 E+04
	MXHOI						
MIDAS		.48 E+03	.44 E+03	.14 E+05	.24 E+03	.23 E+03	.18 E+04
MINV		.75 E+03	.66 E+03	.24 E+05	.36 E+03	.35 E+03	.29 E+04
OMNITAB (INVERT)**		.39 E+03	.36 E+03	.24 E+05	.19 E+03	.18 E+03	.34 E+04
SOLVE	nó IMPRUV						
	IMPRUV	.22 E−02	.12 E−02	.74 E+04	.60 E−03	.36 E−03	.90 E+03

Matrix T^3_{20} X=Approximate inverse $N(A^{-1}-X) \leq \frac{N(XY)}{1-N(Y)} \leq \frac{N(X)N(Y)}{1-N(Y)}$

Condition number .43 E+07 $Y=I-AX$ $\frac{N(A^{-1}-X)}{N(A^{-1})} \leq \frac{N(XY)}{1-N(Y)} \cdot \frac{1+N(Y)}{N(X)}$

$^{**}Y=I-XA$

In conclusion, we would like to make the following observations:

1. Some indication of the possibility of poor results should be included in the output of every program.

2. Some indication of the accuracy of the results should be included in the output of every program.

3. SOLVE with iterative improvement is certainly an excellent program.

4. Interval arithmetic was quite effective but not in every case.

5. The use of $\frac{N(XY)}{1-N(Y)}$ gives a realistic (practical) upper bound on the error of a computed inverse, can be used with either residual matrix and would indicate the possibility of poor results if the norm of each residual matrix were greater than one.

I wish to thank the many people who encouraged me throughout my pursuit of this final step, especially Morris Newman, who made it all possible.

I am grateful to the Lewis and Rosa Strauss Memorial Fund for the financial aid which made the computing facilities of the University of Maryland available to me.

7. References

[1] Fox, L., An Introduction to Numerical Linear Algebra (Oxford Univ. Press, New York, N.Y., 1965), p. 16.
[2] Tompkins, C., and Wilson, W., Jr., Elementary Numerical Analysis (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969), p. 111.
[3] Wampler, R. H., J. Res. Nat. Bur. Stand. (U.S.), **73B** (Math. Sci.), No. 2, 59 (Apr.–June 1969).
[4] Noble, B., Applied Linear Algebra (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969), p. 258.
[5] Belostotskii, A. Y., USSR Comp. Math. Math. Phys. **5**, 151 (1965).
[6] Turing, A. M., Quart J. Mech. Appl. Math. **1**, 301 (1948).
[7] Wilkinson, J. H., Rounding Errors in Algebraic Processes (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963), p. 126.
[8] Albasiny, E. L., Error in digital solution of linear problems, in Error in Digital Computation, Ed. L. B. Rall (John Wiley & Sons, Inc., New York, N.Y., 1965), Vol. 1, p. 171.
[9] Householder, A. S., The Theory of Matrices in Numerical Analysis (Blaisdell Publ. Co., New York, N.Y., 1965), p. 91.
[10] Kahan, W., Can. Math. Bull. **9**, 785 (1966).
[11] Moler, C., J. Assoc. Comp. Mach. **14**, 320 (1967).
[12] Forsythe, G., and Moler, C., Computer Solutions of Linear Algebraic Systems (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967), p. 110.
[13] Forsythe, G., SIAM Rev. **9**, 504 (1967).
[14] Newman, M., and Todd, J., J. SIAM **6**, 467 (1958).

Selected Bibliography

Books

Faddeev, D. K., and Faddeeva, V. N., Computational Methods of Linear Algebra (San Francisco, Calif., W. H. Freeman & Co., 1963).

Forsythe, George, and Moler, Cleve B., Computer Solution of Linear Algebraic Systems (Englewood Cliffs, N.J., Prentice-Hall, Inc., 1967).

Fox, L., An Introduction to Numerical Linear Algebra (New York, N.Y., Oxford Univ. Press, 1965).

Hildebrand, F. B., Introduction to Numerical Analysis (New York, N.Y., McGraw-Hill Book Co., Inc., 1956).

Householder, Alston S., The Theory of Matrices in Numerical Analysis (New York, N.Y., Blaisdell Publ. Co., 1965).

Kelly, Louis G., Handbook of Numerical Methods and Applications (Reading, Penna., Addison-Wesley Publ. Co., 1967).

Noble, Ben, Applied Linear Algebra (Englewood Cliffs, N.J., Prentice-Hall, Inc., 1969).

Perlis, Sam, Theory of Matrices (Cambridge, Mass., Addison-Wesley Publ. Co., Inc., 1952).

Rall, L. B. (ed.), Error in Digital Computation, vol. I (New York, N.Y., John Wiley & Sons, Inc., 1965).

Ralston, Anthony, and Wilf, Herbert S., Mathematical Methods for Digital Computers, vol. I (New York, N.Y., John Wiley & Sons, Inc., 1962).

Ralston, Anthony, and Wilf, Herbert S., Mathematical Methods for Digital Computers, vol. II (New York, N.Y., John Wiley & Sons, Inc., 1967).

Todd, John (ed.), Survey of Numerical Analysis (New York, N.Y., McGraw-Hill Book Co., Inc., 1962).

Tompkins, Charles B., and Wilson, Walter L., Elementary Numerical Analysis (Englewood Cliffs, N.J., Prentice-Hall, Inc., 1969).

Westlake, Joan R., A Handbook of Numerical Matrix Inversion and Solution of Linear Equations (New York, N.Y., John Wiley & Sons, Inc., 1968).

Wilkinson, J. H., Rounding Errors in Algebraic Processes (Englewood Cliffs, N.J., Prentice-Hall, Inc., 1963).

Wilkinson, J. H., The Algebraic Eigenvalue Problem (Oxford, England, Clarendon Press, 1965).

Articles

Albasiny, E. L., Error in digital solution of linear problems, Error in Digital Computation, ed. L. B. Rall (John Wiley & Sons, Inc., New York, 1965), **1**, 131–184.

Ashenhurst, R. L., Techniques for automatic error monitoring and control, Error in Digital Computation, ed. L. B. Rall (John Wiley & Sons, Inc., New York, 1965), **1**, 43–60.

Ashenhurst, R. L., and Metropolis, N., Unnormalized floating point, J. ACM **6**, No. 3, 415–428 (July 1959).

Bauer, F. L., Elimination with weighted row combinations for solving linear equations and least squares problems, Numer. Math. **7**, No. 4, 338–352 (1965).

Belostotskii, A. Y., Estimation of the quality of approximate solutions of a system of linear algebraic equations, USSR Comp. Math. Math. Phys. **5**, No. 1, 151–154 (1965).

Chartres, B., Automatically controlled precision calculations, J. ACM **13**, 386–403 (July 1966).

Chartres, B., and Geuder, J., Computable error bounds for direct solution of linear equations, J. ACM **14**, No. 1, 63–71 (Jan. 1967).

Descloux, J., Note on round-off errors in iterative processes, Math. Comp. **17**, No. 81, 18–27 (Jan. 1963).

Forsythe, G., Today's computational methods of linear algebra, SIAM Review **9**, No. 3, 489–515 (July 1967).

Golstein, M., Significance arithmetic on a digital computer, Comm. ACM **6**, 111–117 (1963).

Gray, H., and Harrison, C., Jr., Normalized floating point with index of significance, Proc. Eastern Joint Computer Conf. (1959), 244–248.

Kahan, W., Numerical linear algebra, Can. Math. Bull. **9**, No. 6, 757–798 (1966).

Lietzke, M. H., Stroughton, R. W., Lietzke, M. P., A comparison of several methods for inverting large symmetric positive definite matrices, Math. Comp. **18**, No. 87, 449–456 (July 1964).

Loizou, G., An empirical estimate of the relative error of the computed solution $\bar{x}$ of $Ax=b$, Comp. J. **11**, No. 1, 91–94 (May 1968).

Longley, J., An appraisal of least squares programs for the electronic computer from the point of view of the user, J. AM. Stat. Assoc. **62**, No. 319, 819–841 (Sept. 1967).

Moler, C., Iterative refinement in floating point, J. ACM **14**, No. 2, 316–321 (Apr. 1967).

Moore, R., The automatic analysis and control of error in digital computing based on the use of interval numbers, Error in Digital Computation, ed. L. B. Rall (John Wiley & Sons, Inc., New York, 1965), **1**, 61–130.

von Neumann, J., and Goldstine, H., Numerical inverting of matrices of high order, Bull. Am. Math. Soc. **53**, 1021–1099 (1947).

Newman, Morris, Matrix computations, Survey of Numerical Analysis, ed. John Todd (McGraw-Hill Book Co., Inc., New York, 1962), 222–254.

Newman, Morris, Solving equations exactly, J. Res. Nat. Bur. Stand. (U.S.), **71B** (Math. & Math. Phys.), No. 4, 171–179 (Oct.–Dec. 1967).

Newman, M., and Todd, J., The evaluation of matrix inversion programs, J. SIAM **6**, No. 4, 466–476 (Dec. 1958).

Oettli, W., and Prager, W., Compatibility of approximate solutions of linear equations with given error bounds for the coefficients and right hand sides, Numer. Math. **6**, No. 5, 405–409 (1964).

Shroeder, J., Computing error bounds in solving linear systems, Math. Comp. **16**, No. 79, 323–337 (July 1962).

Todd, J., The problem of error in digital computation, Error in Digital Computation, ed. L. B. Rall (John Wiley & Sons, Inc., New York, 1965), **1**, 3–42.

Turing, A. M., Rounding-off errors in matrix processes, Quart. J. Mech. Appl. Math. **1**, 287–308 (1948).

Wampler, R., An evaluation of linear least squares computer programs, J. Res. Nat. Bur. Stand. (U.S.), **73B** (Math. Sci.), No. 2, 59–90 (Apr.–June 1969).

(Paper No. 74B4–336)