NUMERICAL LINEAR ALGEBRA

W. Kahan

This is the 1st half of a lecture delivered before the Canadian Math. Congress in Québec in Sept. 1965.

Reprinted from
Canadian Mathematical Bulletin, Volume 9, No. 6, 1966

Corrected,

NUMERICAL LINEAR ALGEBRA

W. Kahan

Table of Contents

0. Introduction. The primordial problems of linear algebra are the solution of a system of linear equations

$$A\underline{x} = \underline{b} \quad (\text{i.e., } \Sigma_j a_{ij} x_j = b_i) \text{ ,}$$

and the solution of the eigenvalue problem

$$A\underline{x}_k = \lambda_k \underline{x}_k$$

for the eigenvalues λ_k and corresponding eigenvectors $\underline{x}_k$ of a given matrix A . The numerical solution of these problems without the aid of an electronic computer is a project not to be undertaken lightly. For example, using a mechanical desk-calculator to solve five linear equations in five unknowns (and check them) takes me nearly an hour, and to calculate five eigenvalues and eigenvectors of a five-by-five matrix costs me at least an afternoon of drudgery. But any of today's electronic computers are capable of performing both calculations in less than a second.

* Sections 6 to 11 will appear in this Bulletin at a later date.

What is the current state of the art of solving numerical problems in linear algebra with the aid of electronic computers? That question is the theme of part of this paper. The rest of the paper touches upon two or three of the collateral mathematical problems which have captured my attention for the past several years. These problems spring from the widespread desire to give the computer all its instructions in advance. When the computer is engrossed in its computation at the rate of perhaps a million arithmetic operations per second, human supervision is at best superficial. One dare not interrupt the machine too frequently with requests "WHAT ARE YOU DOING NOW?" and with afterthoughts and countermands, lest the machine be dragged back to the pace at which a human can plod through a morass of numerical data. Instead, it is more profitable to launch the computer on its phrenetic way while we calmly embrace mathematical (not necessarily computational) techniques like error-analysis to predict and appraise the machine's work. Besides, the mathematical problems of prediction and appraisal are interesting in their own right.

1. The Time Needed to Solve Linear Equations. On our computer (an IBM 7094-II at the University of Toronto) the solution of 100 linear equations in 100 unknowns can be calculated in about 7 seconds; during this time the computer executes about 5000 divisions, 330000 multiplications and additions, and a comparable amount of extra arithmetic with small integers which ensures that the aforementioned operations are performed in their correct order. This calculation costs about a dollar. To calculate the inverse of the coefficient matrix costs about three times as much. If the coefficients are complex instead of real, the cost is roughly quadrupled. If the same problem were taken to any other appropriate electronic computer on this continent, the time taken could differ by a factor between 1/10 and 1000 (i.e. 1 second to 2 hours for 100 equations) depending upon the speed of the particular machine used. These quotations do not include the time required to produce the equations' coefficients in storage, nor the time required (a few seconds) to print the answers on paper.

For the next five years it will be economically practical to solve general systems of 1000 or so linear equations, but not 10000. One limitation is the need to store the equations' coefficients somewhere easily accessible to the computer's arithmetic units. A general system of N equations has an NxN coefficient matrix containing N^2 elements. When N=100 , these 10000 elements fit with ease into current storage units. When N=1000 , finding the space for a million elements requires today some attention to technical details; tomorrow's storage units will handle a million elements easily. But when N=10000 space is needed for 10^8 elements, and current storage units with that capacity are unable to share their information with the computer at speeds commensurate with its arithmetic units. Besides, to produce, collect, and check those 10^8 elements is a formidable undertaking.

Today, the solution of 1000 equations is not a simple task, even on a large computer like ours. Our computer's immediate access store, to which reference can be made in a fraction of the time required for one multiplication, has a capacity of 2^{15}=32768 words, of which about 10000 would be needed for program. The remaining space is just about enough for 10 or 20 rows of the matrix. The rest of the matrix, 980 rows, has to be kept in bulk storage units, like magnetic tapes or disks, to which access takes at least as long as several multiplications. Now, most of the time spent in solving a linear system is spent thus:

> Select an element from the matrix,
> multiply by another,
> subtract the product from a third,
> which is then replaced by the difference.

It is clear that careful organization is required to prevent storage-access from consuming far more time than the arithmetic. Such organization is possible; for a good example see Barron and Swinnerton-Dyer (1960). The main idea is to transfer each row of the matrix in turn from slow storage to fast storage and back in such a way that, while in fast storage, each row partakes in as many arithmetic operations with neighbouring rows as possible. Further time is saved by the simultaneous execution of input, output and calculation; while one row is being transferred from slow to fast storage, another is being transferred back, and arithmetic operations are being performed upon a third row. In this way, 1000 linear equations could be solved on our machine in a morning, not much longer than would be needed for the arithmetic operations alone (and in much less time than would likely be needed to collect the data or to interpret the answer).

Let us count up those arithmetic operations. The methods most widely used for solving linear equations are elimination methods patterned after that described by Gauss (1826). Here is an outline:

Given the augmented matrix $\{A,\underline{b}\}$ of the system

$$\sum_{j=1}^{N} a_{ij}x_j = b_i \ , \qquad i=1,2,\ldots,N \ ,$$

we select a variable, say x_J , and eliminate it from all the equations but one. This can be done, for example, by selecting a suitable equation, say the I^{th} , and subtracting (the I^{th} equation, times a_{iJ}/a_{IJ}) from (the i^{th} equation) for all $i \neq I$. After the I^{th} equation and J^{th} variable have been set aside, one has just (N-1) linear equations in (N-1) unknowns left.

This simple process is repeated until there remains only one equation in one unknown; this can be solved easily. The solution is substituted back into the equation previously set aside and that one is solved. This process of back substitution is repeated until, at the end, x_J is obtained from the I^{th} equation after the substitution of the computed values of the other N-1 variables.

Gaussian elimination requires

$$\tfrac{1}{3}N^3(1 + \tfrac{3}{2}N^{-1} - \tfrac{5}{2}N^{-2}) \text{ additions or subtractions,}$$

and

$$\tfrac{1}{3}N^3(1 + 3N^{-1} - N^{-2}) \text{ multiplications or divisions.}$$

In the 140 years since this method appeared in print, many other methods, some of them very different, have been proposed. All methods satisfy the following new theorem of Klyuyev and Kokovkin-Shcherbak (1965):

> Any algorithm which uses only rational "row & column" arithmetic operations to solve a general system of N linear equations requires at least as many additions and subtractions, and at least as many multiplications and divisions as does Gaussian elimination. X

One consequence of this theorem is obtained by setting N=10000; to solve 10000 linear equations would take more than two months for the arithmetic alone on our machine. The time might come down to a day or so when machines 100 times faster than ours are produced, but such machines are just now being developed, and are most unlikely to be in widespread use within the next five years. The main impediment seems to be storage access time. (For more details, see IFIP (1965).)

In the meantime, there are several good reasons to want to solve systems of as many as 10000 equations. For example, the study of many physical processes (radiation, diffusion, elasticity,...) revolves about the solution of partial differential equations. A powerful technique for solving these differential equations is to

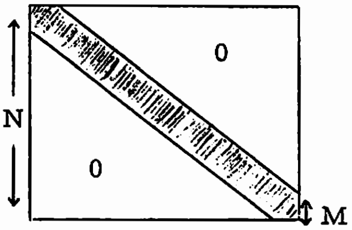
X
S. Winograd has shown how to nearly halve the work, though at the cost of a possible loss in numerical stability.
See IFIP 68

r. Strassen has brought the work down to $\propto N^{\log_2 7}$.

approximate them by difference equations over a lattice erected to approximate the continuum. The finer the lattice (i.e. the more points there are in the lattice), the better the approximation. In a 20x20x20 cubic lattice there are 8000 points. To each point corresponds an unknown and a difference equation. Fortunately, these equations have special properties which free us from the limitation given by Klyuyev and Kokovkin-Shcherbak. (For details about partial difference equations, see Smith (1965) or Forsythe and Wasow (1960).)

2. <u>The Time Needed to Solve a Linear System with a Band Matrix.</u> The systems of linear equations which arise from the discretization of boundary value problems frequently have matrices $\{a_{ij}\}$ with the following "band property":

$$a_{ij} = 0 \quad \text{if} \quad |i-j| > M .$$



<u>A Band-Matrix.</u>

Although N equals the number of lattice points in the discretization, and can therefore be quite large whenever a fine lattice is needed for high accuracy, the half-bandwidth M is usually much smaller than N . For a boundary value problem in δ dimensions M is usually very near the number of points in one or two (δ-1) dimensional sections of the lattice, and hence the quotient

$$M/N^{1-1/\delta}$$

is frequently between 1 and 3 . With care, the matrix corresponding to k coupled boundary value problems over the same lattice can often be put in a band form for which the quotient above lies between k and 3k .

The advantage of a band structure derives from the fact that it is preserved by the row-operations involved in Gaussian elimination. This is obvious when we select, for $I = 1,2,\ldots, N$ in turn, the I^{th} equation to eliminate the I^{th} unknown from all subsequent equations. It is true also when any other row-selection rule is used, provided the width of that part of the band above the main diagonal is allowed to increase by M. Consequently, far less time and space are needed to solve band-systems than to solve general systems. The following table summarizes the dependence of time and space requirements upon the parameters M and N. For the sake of simplicity, constants of proportionality have been omitted, and terms of the order of $1/M$ and $1/N$ or smaller are ignored.

Type of Matrix	Storage required (Total)	Time required for arithmetic alone
Band	MN	M^2N
Full	N^2	N^3

Incidentally, there is no need to find space for all MN elements of the band-matrix in the immediate-access store. Instead, it suffices to store the rows of the matrix in slow bulk storage (tape or disk) and find space for $2M^2$ elements (in some cases fewer) in the immediate access store. Then as the I^{th} row of the matrix is transferred from slow to fast storage, a transformed $(I-M)^{th}$ row can be transferred from fast to slow.

The economies that result from band structure permit the solution of two dimensional boundary value problems with thousands of points, and one dimensional problems with tens of thousands of points, to be carried out in times measured in minutes. (For more details about the solution of band systems, see Cayless (1961), Fox (1957) pp. 150-155, Walsh's ch. 22 in Fox (1962), or Varga (1962) pp. 194-201.)

3. Iterative Methods for Solving Linear Systems. Iterative methods for solving linear systems, due originally to Gauss (1823), Liouville (1837) and Jacobi (1845), embody an approach which is very different from that behind the direct methods like Gaussian elimination. The difference can be characterized as follows:

Direct methods apply to the equation $A\underline{x} = \underline{b}$ a finite sequence of transformations at whose termination the equations have a new form, say $U\underline{x} = \underline{c}$, which can be solved by an obvious and finite calculation. For example, in Gaussian elimination U is an upper triangular matrix which, with $\underline{c}$, can be shown to satisfy

$$\{U,\underline{c}\} = L^{-1}P\{A,\underline{b}\}$$

where P is a permutation of the identity and L is a lower triangular matrix with diagonal elements all equal to 1 ; and the obvious calculation that solves $U\underline{x} = \underline{c}$ is back substitution. In the absence of rounding errors, the computed solution is exact. (For more details see Faddeev and Faddeeva (1964) ch. II, Householder (1964) ch. 5, or Fox (1964) ch. 3-5.)

On the other hand, an iterative method for solving $A\underline{x} = \underline{b}$ begins with a first approximation $\underline{z}_0$, to which a sequence of transformations are applied to produce a sequence of iterates $\underline{z}_1$, $\underline{z}_2$, $\underline{z}_3$,... which are supposed to converge toward the desired solution $\underline{x}$. In practice the sequence is terminated as soon as some member $\underline{z}_k$ of the sequence is judged to be close enough to $\underline{x}$.

An example of an iterative method is the Liouville-Neumann series which is produced by what numerical analysts call "Jacobi's Method":

Suppose $A\underline{x} = \underline{b}$ can be transformed conveniently into the form

$$\underline{x} = B\underline{x} + \underline{c}$$

where the matrix B is small in some sense. To be more precise, we shall assume that $\|B\| = \beta < 1$. (The symbol $\|...\|$ represents a matrix norm about which more will be said later.) We begin with a first approximation $\underline{z}_0$, for which $\underline{0}$ will do if nothing better is known, and iterate thus:

$$\begin{aligned} \underline{z}_{k+1} &= B\underline{z}_k + \underline{c} \quad \text{for} \quad k = 0,1,2,\dots , \\ &= (I+B+B^2+ \dots +B^k)\underline{c} + B^{k+1}\underline{z}_0 \\ &= \underline{x} + B^{k+1}(\underline{z}_0-\underline{x}) \ . \end{aligned}$$

Hence, the error is bounded by

$$\|\underline{z}_k-\underline{x}\| < \beta^k \|\underline{z}_0-\underline{x}\| \ .$$

The practicality of this scheme depends upon three considerations:

i) The smaller is β , the fewer are the iterations required to effect a given factor of reduction in the error bound. Therefore, small values of β are desired for rapid convergence.

ii) The better is $\underline{z}_0$, the fewer are the iterations required to bring the error bound below a given tolerance. Therefore, good initial approximations are desired for early termination of the iteration.

iii) If each matrix-vector multiplication $B\underline{z}_k$ is cheap enough that we can afford a large number of them, then the two previous considerations will carry less weight.

The last consideration is quite important in many applications. For example, if the NxN matrix B is "sparse", which means that most of B's elements are zeros, then the number of arithmetic operations required to compute $B\underline{z}_k$ may well be a small multiple of N instead of N^2 . Such sparse matrices are frequently encountered during the study of trusswork bridges, electric networks, economists' input-output models, and boundary value problems. In the case of some large two-dimensional boundary value problems, and most three dimensional ones, it may be more economical to exploit the sparseness of the matrix than to exploit its band properties.

Despite a sparse matrix and a fast computer, the simple iteration described above is usually too slow to be practical. This fact has spurred the generalization and improvement of iterative methods in a vast diversity of ways.

One generalization begins with the iteration

$$\underline{z}_{k+1} = \underline{z}_k + C_k \underline{r}_k$$

where

$$\underline{r}_k = \underline{b} - A\underline{z}_k$$

is $\underline{z}_k$'s "residual" in the equation $A\underline{x} = \underline{b}$. (The Jacobi iteration is obtained formally by letting C_k^{-1} be A's diagonal.) For simplicity suppose $C_k = \gamma_k C$, with scalars γ_k and the matrix C to be chosen later in accordance with the subsequent analysis. We find that

$$\underline{z}_k = \underline{x} + P_k(CA)(\underline{z}_0 - \underline{x})$$

where $P_k(w)$ is a k^{th} degree polynomial defined by the recurrence

$$P_0(w) = 1 \; , \; P_{k+1}(w) = (1-\gamma_k w)P_k(w) \; .$$

To simplify the exposition now I shall assume that CA's elementary divisors are all linear; otherwise what follows would be complicated by the appearance of some derivatives of $P_k(w)$. The matrix CA can be decomposed into its idempotent elements $E(\lambda)$ defined by

$$(CA)^n = \sum_\lambda \lambda^n E(\lambda) \quad \text{for all} \quad n$$

where the summation is taken over the values of the eigenvalues λ of CA . (Cf. Dunford and Schwartz (1958) pp. 558-9.) Then

$$P_k(CA) = \sum_\lambda P_k(\lambda) E(\lambda) \; ,$$

whence comes the following theorem:

A necessary and sufficient condition that $\underline{z}_k \rightarrow \underline{x}$ as $k \rightarrow \infty$, no matter how $\underline{z}_0$

be chosen, is that $P_k(\lambda) \to 0$ for every eigenvalue λ of CA .

This theorem is not applied directly in practice because the eigenvalues λ are generally not known. But if all the eigenvalues λ are known to be contained in some region R in the complex plane, then it suffices that $P_k(w) \to 0$ for all w in R . To satisfy this requirement is not trivial, because $P_k(0) = 1$ for all k .

Given R and k , one might seek a polynomial $P_k(w)$ which is "best" in the sense that, for example, of all polynomials of degree k for which $P_k(0) = 1$, $P_k(w)$ has the smallest value of

$$\max |P_k(w)| \quad \text{over} \quad w \quad \text{in} \quad R .$$

There is no general rule known for finding such best polynomials. The following theorem helps in some cases:

Let $\mathcal{L}_k(r)$ be the lemniscate in the complex w-plane defined by

$$|L_k(w)| \leq r < L_k(0)$$

for some polynomial $L_k(w)$ of degree $\geq k$.

Then every k^{th} degree polynomial $P_k(w)$ with $P_k(0) = 1$ satisfies

$$\max |P_k(w)| \geq r/L_k(0) \quad \text{for} \quad w \quad \text{in} \ \mathcal{L}_k(r) .$$

Proof. Apply the maximum modulus theorem on the exterior of $\mathcal{L}_k(r)$ to the rational function

$$L_k(0)\ P_k(w)/L_k(w)$$

to conclude that this quotient has magnitude at least 1 somewhere on the boundary of $\mathcal{L}_k(r)$. Then apply the same theorem to $P_k(w)$ inside $\mathcal{L}_k(r)$.

The simplest application of this theorem is to the circle

$$|L_k(w)| = |(1-w)^k| \le \beta^k \quad ,$$

which shows that, if $A = I-B$ and we know only that $\|B\| = \beta < 1$ (so that all eigenvalues λ of A must satisfy $|1-\lambda| \le \beta < 1$), then the simple Jacobi iteration described above is the best that can be done.

Having chosen a polynomial $P_k(w)$, the numbers γ_j are defined as the reciprocals of the zeros of $P_k(w)$. This relation amounts to an inconvenient restriction on the sequence of polynomials $P_j(w)$ for $j < k$, and is also a source of possible numerical instability. To illustrate this point, suppose all the eigenvalues λ of CA lie in an interval on the real axis, say $0 < \alpha_0 \le \lambda \le \alpha_m$. The following theorem of Markoff is applicable:

> Of all k^{th} degree polynomials $P_k(z)$ with $P_k(0) = 1$, the one for which
>
> $$\max |P_k(z)| \quad \text{on} \quad \alpha_0 \le z \le \alpha_m$$
>
> is smallest is just the Tchebycheff polynomial
>
> $$T_k(L(z))/T_k(L(0))$$
>
> where
>
> $$T_k(\cos\theta) = \cos k\theta \quad \text{and}$$
>
> $$L(z) = (\alpha_m + \alpha_0 - 2z)/(\alpha_m - \alpha_0) \ .$$

Proof. If $|P_k(z)| \le 1/T_k(L(0))$ on $\alpha_0 \le z \le \alpha_m$, then the difference $T_k(L(z)) - T_k(L(0))P_k(z)$ has at least one zero between each pair of adjacent extrema of $T_k(L(z))$ on the interval, and another at $z = 0$, making $k+1$ zeros altogether.

Now, the zeros γ_j^{-1} of $T_k(L(z))$ include some numbers quite near α_0, which may be very small in cases of slow convergence. But then the term $\gamma_j C\underline{r}_j$, when its turn comes in the iteration, can be so large and so much magnify the effects of rounding errors that the convergence of the iteration is jeopardized (Young (1956)).

Fortunately, the Tchebycheff polynomials satisfy a three-term recurrence

$$T_{k+1}(L(z)) = 2L(z)T_k(L(z)) - T_{k-1}(L(z))$$

which can be implemented conveniently and is numerically stable. A suitable form for the iteration is

$$\underline{z}_{k+1} = \underline{z}_k + \gamma_k C\underline{r}_k + \delta_k(\underline{z}_k - \underline{z}_{k-1}) ,$$

and an appropriate choice of γ_k, δ_k, and $\underline{z}_{-1} = \underline{z}_0$, yields

$$\underline{z}_k = \underline{x} + T_k(L(CA))(\underline{z}_0 - \underline{x})/T_k(L(0))$$

for all $k \geq 0$. One great convenience of this iteration is that the polynomial $P_k(w)$ that appears in the relation

$$\underline{z}_k = \underline{x} + P_k(CA)\ (\underline{z}_0 - \underline{x})$$

is the "best" such polynomial for each k, so that there is no need to choose a degree k in advance. This convenience persists whenever the sequence of polynomials $P_k(w)$ are orthogonal polynomials which minimize some weighted mean value of $|P(w)|^2$ over an interval of interest, because these polynomials also satisfy a three-term recurrence. For details see Stiefel (1958) and Faddeev and Faddeeva (1964) ch. 9.

A very different approach to iterative methods can be illustrated by the Jacobi method once again. We note that, since

$$\underline{z}_{k+1} - \underline{x} = B(\underline{z}_k - \underline{x}) ,$$

$$\|\underline{z}_{k+1} - \underline{x}\| \leq \beta \, \|\underline{z}_k - \underline{x}\|$$

where $\beta = \|B\| < 1$. In other words, each iteration reduces the norm of the error by a factor $\beta < 1$.

More generally, given a norm for the error $\|\underline{z}_k - x\|$ or for the residual $\|\underline{b} - A\underline{z}_k\|$, one seeks a direction $\underline{p}_k$ and distance γ_k such that the norm associated with

$$\underline{z}_{k+1} = \underline{z}_k + \gamma_k \underline{p}_k$$

is smaller than for $\underline{z}_k$. In Gauss-Seidel iterations the direction $\underline{p}_k$ is one of the coordinate directions; in gradient iterations the direction $\underline{p}_k$ is that in which the norm is decreasing most quickly. For further discussion see Householder (1964) sec. 4.2-3. I shall elaborate upon only one such method, called "the method of conjugate gradients".

Suppose A is symmetric and positive definite, and let us use $\|\underline{r}_k\| = (\underline{r}_k{}^\tau \underline{r}_k)^{1/2}$ as a norm for the residual $\underline{r}_k = \underline{b} - A\underline{z}_k$. Then each iteration step

$$\underline{z}_{k+1} = \underline{z}_k + \gamma_k \underline{r}_k + \delta_k(\underline{z}_k - \underline{z}_{k-1})$$

looks formally just like the iteration that was used above to construct the Tchebycheff polynomials, but now the constants γ_k and δ_k must be chosen to minimize $\|\underline{r}_{k+1}\|$ in that step. This choice of γ_k and δ_k has the stronger property that no other choice of $\gamma_0, \delta_0, \gamma_1, \delta_1, \ldots, \gamma_k, \delta_k$ could yield a smaller value for $\|\underline{r}_{k+1}\|$! In particular, $\underline{r}_N = \underline{0}$; the iteration converges in a finite number of steps. An excellent exposition of this powerful technique is given by Stiefel (1953) and (1958).

Another approach to iterative methods is embodied in the relaxation methods. The basic idea here is to adjust some unknown x_J to satisfy ("relax") the I^{th} equation $\Sigma_j a_{Ij} x_j = b_i$, even though in doing so some other equation may be dissatisfied. The next step is to relax some other equation, and so on. Gauss (1823) claimed that this iteration could be performed successfully

> "... while half asleep, or while thinking about other things".

Since his time the method has been systematized and generalized and improved by orders of magnitude, especially where its applications to discretized boundary value problems are concerned. The best survey of this development is currently to be found in Varga's book (1962). Nowadays some of the most active research into iterative methods is being conducted upon those variants of relaxation known as Alternating Direction Methods; see for example Douglas and Gunn (1965), Gunn (1965), Murray and Lynn (1965), and Kellog and Spañier (1965).

The result of the past fifteen years of intense mathematical analysis concentrated upon iterative methods has been to speed them up by factors of ten and a hundred. Some idea of the times involved can be gained from surveys by Engeli et al (1959), Martin and Tee (1961), and Birkhoff, Varga and Young (1962). For example, the difference analogue of Dirichlet's problem (Laplace's equation with specified boundary values) in a two-dimensional region with about 3000 lattice points can be solved to within about 6 significant decimals in about 300 iterations of successive over-relaxation, requiring about 30 seconds on our machine. This is one third as long as would be needed to apply Gaussian elimination to the corresponding band matrix. A three-dimensional problem with 10000 equations and unknowns could be solved on our machine in less than 5 minutes by iteration; here Gaussian elimination takes hundreds of times as long, so the value of iteration is well established. But the time required for iterative methods generally cannot easily be predicted in advance except in special cases (which are fortunately not uncommon). Furthermore, the choice of one out of several possible iterative methods is frequently a matter of trial and error. Even if the iterative method is chosen on rational grounds, there will be parameters (like the constants γ_k and δ_k above) which must be chosen carefully for maximum efficiency; but to choose their values will frequently require information that is harder to obtain than the solution being sought in the first place. (A welcome exception is the method of conjugate gradients.) Evidently there is plenty of room for more research, and especially for the consolidation of available knowledge.

4. Errors in the Solution of Linear Systems. It is possible in principle to perform a variant of Gaussian elimination using integer arithmetic throughout in such a way that no rounding errors are committed (see Aitken's method described by Fox (1964) on pp. 82 - 86). But the integers can grow quite long, as much as N times as long as they were to begin with in the given NxN matrix A. Whenever N is large, one is easily persuaded to acquiesce when the computer rounds its arithmetic operations to some finite number of significant digits specified in advance. Consequently, it comes as no surprise when the calculated value $\underline{z}$ of the solution of

$$A\underline{x} = \underline{b}$$

exhibits a small residual

$$\underline{r} = \underline{b} - A\underline{z} .$$

How small must $\underline{r}$ be to be negligible? The following example shows that this question can have a surprising answer.

Example 1.

$$A = \begin{pmatrix} .2161 & .1441 \\ 1.2969 & .8648 \end{pmatrix}, \quad \underline{b} = \begin{pmatrix} .1440 \\ .8642 \end{pmatrix}, \quad \underline{z} = \begin{pmatrix} .9911 \\ -.4870 \end{pmatrix} .$$

Then the residual is

$$\underline{r} = \underline{b} - A\underline{z} = \begin{pmatrix} -.00000001 \\ .00000001 \end{pmatrix} \text{ exactly ;}$$

no other vector $\underline{z}$ specified to 4 dec. can have a smaller residual $\underline{r}$ unless $\underline{r} = \underline{0}$. But $\underline{z}$ does not contain a single correct digit! The"correct" solution is

$$\underline{x} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} .$$

Linear systems with this kind of pathological behaviour are often called "ill-conditioned". Precisely what does "ill-conditioned" mean?

This example and other problems of error analysis are easier to discuss using the language of matrix norms, which I digress to introduce here.

A common vector norm is the Hölder norm

$$\|\underline{x}\|_p \equiv (\Sigma_j |x_j|^p)^{1/p} \quad \text{for} \quad 1 \leq p \leq \infty \ .$$

This norm is easily shown to have the properties that one expects of a vector norm:

$$\|\underline{x}\| > 0 \quad \text{except that } \|\underline{0}\| = 0 \ .$$
$$\|\alpha\underline{x}\| = |\alpha| \ \|\underline{x}\| \text{ for all scalars } \alpha \ .$$
$$\|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\| \text{ (The Triangle Inequality) } .$$

Any linear transformation A from one normed linear space to another can be normed in the natural way:

$$\|A\| \equiv \max \|A\underline{x}\| / \|\underline{x}\| \text{ over } \underline{x} \neq \underline{0} \ .$$

(I use "max" instead of "sup" because they amount to the same thing for finite dimensional spaces.) Among the matrix norms most often used are

$\|A\|_{p,q} \equiv \max \|A\underline{x}\|_p / \|\underline{x}\|_q$ with suitably chosen p and q ; e.g.

$$\|A\|_{\infty,\infty} = \max \|A\underline{x}\|_\infty / \|\underline{x}\|_\infty = \max_i \cancel{|}\Sigma_j |a_{ij}| \text{ (max row-sum)}$$
$$\|A\|_{1,1} = \max \|A\underline{x}\|_1 / \|\underline{x}\|_1 = \max_j \cancel{|}\Sigma_i |a_{ij}| \text{ (max column-sum)}$$
$$\|A\|_{\infty,1} = \max \|A\underline{x}\|_\infty / \|\underline{x}\|_1 = \max_{ij} |a_{ij}|$$
$$\|A\|_{2,2} = \max \|A\underline{x}\|_2 / \|\underline{x}\|_2 = (\text{max eigenvalue of } A^\tau A)^{1/2}$$

(Another matrix norm, $\|A\|_E \equiv (\text{trace } A^\tau A)^{1/2}$, is widely used but cannot be defined as a maximum of a ratio of two vector norms. Its main value is as an easily computed estimate of $\|A\|_{2,2}$, because

$$\|A\|_E \geq \|A\|_{2,2} \geq \|A\|_E / \sqrt{\text{rank } (A)} \ . \ )$$

Until further notice, the matrix norm used below will be assumed to be one of the norms $\|A\|_{p,q}$.

Finally, the notion of a dual linear functional should be mentioned.. If the row-vector $\underline{y}^\tau$ is regarded as a linear operator from a normed vector space to the space of complex numbers normed as usual, then

$$\|\underline{y}^\tau\| = \max |\underline{y}^\tau\underline{x}|/\|\underline{x}\| \quad \text{over} \quad \underline{x} \neq \underline{0} \ .$$

e.g. $\|\underline{y}^\tau\|_p = \max |\underline{y}^\tau\underline{x}|/\|\underline{x}\|_p = \|\underline{y}\|_q$ where $p^{-1} + q^{-1} = 1$.

And the Hahn-Banach theorem guarantees that to each $\underline{x} \neq 0$ corresponds a dual functional $\underline{y}^\tau$ such that

$$\underline{y}^\tau\underline{x} = \|\underline{y}^\tau\| \ \|\underline{x}\| = 1 \ .$$

(For more details, see Householder (1964) or any text on normed linear spaces; e.g. Day (1962), Kantorovich and Akilov (1964).)

Now it is possible to discuss the meaning of "ill-condition". To each matrix A, regarded as a linear operator from one normed space to another, can be assigned its condition number $K(A)$ associated with the norms and defined thus:

$$K(A) \equiv \frac{\max \|A\underline{x}\|/\|\underline{x}\|}{\min \|A\underline{y}\|/\|\underline{y}\|} \quad \text{over all } \underline{x} \neq \underline{0} \text{ and } \underline{y} \neq \underline{0} \ .$$

In other words, if the vectors $\Delta\underline{x}$ and $\Delta\underline{b}$ are regarded as errors correlated by $A(\underline{x} + \Delta\underline{x}) = \underline{b} + \Delta\underline{b}$, where $\underline{x}$ and $\underline{b}$ satisfy $A\underline{x} = \underline{b}$, then

$$1/K(A) \leq (\|\Delta\underline{x}\|/\|\underline{x}\|)/(\|\Delta\underline{b}\|/\|\underline{b}\|) \leq K(A) \ .$$

This means that a small change $\Delta\underline{b}$ in $\underline{b}$ causes a change $\Delta\underline{x}$ in $\underline{x}$ which has, relatively speaking, a norm that can be $K(A)$ times as big. When $K(A)$ is very large, we say that "A is ill-conditioned".

It is easy to prove that when A is a square matrix,

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

The matrix A of the numerical example is very ill-conditioned indeed;

$$A^{-1} = \begin{pmatrix} -86480000. & 14410000. \\ 129690000. & -21610000. \end{pmatrix} \quad \text{and}$$

$$K(A) \doteqdot 2 \times 10^8 .$$

If we apply the inequality, associating $\underline{r}$ with $\Delta\underline{b}$ and $\underline{z} - \underline{x}$ with $\Delta\underline{x}$, we verify that

$$(\|\underline{z} - \underline{x}\|/\|\underline{x}\|)/(\|\underline{r}\|/\|\underline{b}\|) \doteqdot (1/2)/(10^{-8}) < K(A) .$$

Had $K(A)$ been known in advance, the example would not have come as a surprise.

Another important property of the condition number is given by the following:

THEOREM: A differs from a singular matrix by no more in norm than $\|A\|/K(A)$, (Gastinel) i.e., given A, $\|A\|/K(A) = \min \|\Delta A\|$ over all singular $(A + \Delta A)$.

Proof. Of course, if $(A + \Delta A)$ is singular, then there is some $\underline{x} \neq \underline{0}$ for which $(A + \Delta A)\,\underline{x} = \underline{0}$.

$$\begin{aligned} \text{Therefore } \|\Delta A\| &\geq \|\Delta A\underline{x}\|/\|\underline{x}\| \\ &= \|A\underline{x}\|/\|\underline{x}\| \\ &= \|A\underline{x}\|/\|A^{-1}\ A\underline{x}\| \\ &\geq 1/\|A^{-1}\| = \|A\|/K(A) . \end{aligned}$$

To find a ΔA which achieves the inequality we consider that vector $\underline{y}$ for which
$\|A^{-1}\ \underline{y}\| = \|A^{-1}\|\ \|\underline{y}\| \neq 0$.
Then let $\underline{w}^\tau$ be dual to $A^{-1}\ \underline{y}$

$$\text{i.e. } \underline{w}^\tau\ A^{-1}\ \underline{y} = \|\underline{w}^\tau\|\ \|A^{-1}\ \underline{y}\| = 1 ,$$

and set $\Delta A = -\ \underline{y}\ \underline{w}^\tau$.

We have $(A + \Delta A)\ A^{-1}\ \underline{y} = 0$, so $A + \Delta A$ is singular.

And

$$\begin{aligned} \|\Delta A\| &= \max \|\underline{y}\ \underline{w}^\tau\underline{x}\|/\|\underline{x}\| \quad \text{over } \underline{x} \neq \underline{0} \\ &= \|\underline{y}\| \max |\underline{w}^\tau\underline{x}|/\|\underline{x}\| \end{aligned}$$

$$= \|\underline{y}\| \; \|\underline{w}^\tau\| = \|\underline{y}\|/\|A^{-1}\underline{y}\| = 1/\|A^{-1}\|$$
$$= \|A\|/K(A) \; .$$

Let us return to the example again. If the elements of A have been rounded to 4 decimal places, then the uncertainty introduced by the rounding is 10000 times larger than the difference between A and the nearest singular matrix! Under these circumstances it is reasonable to ask whether the system $A\underline{x} = \underline{b}$ deserves to have a solution.

The pathological behaviour of ill-conditioned matrices seems to have preyed upon the minds of the early analysts of the error committed during Gaussian elimination. Certainly the conclusions of von Neumann and Goldstine (1947, 1951) are incredibly pessimistic; for example they concluded that on a machine like ours there were substantial risks taken in the numerical inversion of matrices of orders much larger than 20, although their error-analysis was correct in other respects. (Their trouble arises from an attempt to compute A^{-1} from the formula

$$A^{-1} = (A^\tau A)^{-1} A^\tau \; ,$$

an attempt which we know now to have been ill-conceived.)

A more nearly modern error-analysis was provided by Turing (1948) in a paper whose last few paragraphs foreshadowed much of what was to come, but his paper lay unnoticed for several years until Wilkinson (1960) began to publish the papers which have since become a model of modern error-analysis.

Wilkinson's main result about Gaussian elimination can be summarized thus:

Provided Gaussian elimination is carried out in a reasonable way (about which more later), the computed approximation $\underline{z}$ to the solution of

$$A\underline{x} = \underline{b}$$

will satisfy instead an equation of the form

$$(A + \Delta A)\, \underline{z} = \underline{b}$$

where, although ΔA is not independent of $\underline{b}$ and $\underline{z}$, ΔA satisfies an inequality of the form

$$\|\Delta A\| \leq c \; g_N \; N^p \; \beta^{-s} \; \|A\|$$

where N is the order of A ,

β^{-s} represents "1 unit in the last place" e.g. $\beta^{-s} = 10^{-8}$ on an 8 dec. digit machine,

c is a modest positive constant, usually less than 10,

p is a small positive constant always smaller than 3.

g_N is the pivot-ratio, about which more later.

The constants c and p depend upon the details of the arithmetic and the norm; they will not be discussed here. (See Wilkinson (1963) and references cited therein.)

In short, $\|\Delta A\|$ is comparable to rounding errors in $\|A\|$; and if the data in A is already uncertain by more than a few hundred units in the last place carried then the perturbation ΔA attributable to the process of elimination will be quite negligible by comparison. Indeed, in many cases the perturbation ΔA will amount to less than one unit in the last place of each element of A !

So, a small residual

$$\underline{r} = \underline{b} - A\underline{z} = \Delta A \; \underline{z}$$

is just what might be expected from Gaussian elimination. But the error $\underline{z} - \underline{x}$ is another matter;

$$\underline{z} - \underline{x} = -A^{-1} \; \Delta A \; \underline{z} \; , \quad \text{where } \|\Delta A\| \leq \varepsilon \; \|A\| \; , \text{ say .}$$

Therefore $\|\underline{z} - \underline{x}\| \leq \|A^{-1}\| \; \|\Delta A\| \; \|\underline{z}\|$

$$\leq \frac{K(A) \; \varepsilon}{1 - K(A) \; \varepsilon} \; \|\underline{x}\|$$

where K(A) is A's condition number,

$\varepsilon = c \; g_N \; N^p \; \beta^{-s}$ and is very small,

and we assume that $K(A) \; \varepsilon < 1$.

In other words, although the residual $\underline{r}$ is always small, the error $\underline{z} - \underline{x}$ may be very large if A is ill-conditioned; but this error will be no worse than if A were in error by about $c\ g_N\ N^p$ units in its last place to begin with.

The constant g_N has an interesting history. It is connected with the rate of growth of the "pivots" in the Gaussian elimination scheme. The pivot is the coefficient a_{IJ} by which the I^{th} equation is divided during the elimination of x_J . This term "pivot" will be easier to explain after the following example has been considered:

Example 2.

$$A = \begin{array}{cccc} \underline{x_1} & \underline{x_2} & \underline{x_3} & \\ \left(\begin{array}{c} 2 \times 10^{-10} \\ -1 \\ 1 \end{array}\right. & \begin{array}{c} -1 \\ 1 \\ 1 \end{array} & \left.\begin{array}{c} 1 \\ 1 \\ 1 \end{array}\right) & \begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array} \end{array} ,$$

$$A^{-1} = \tfrac{1}{4} \begin{array}{cccc} \underline{b_1} & \underline{b_2} & \underline{b_3} & \\ \left(\begin{array}{c} 0 \\ -2 \\ 2 \end{array}\right. & \begin{array}{c} -2 \\ 0.9999999998 \\ 1.0000000002 \end{array} & \left.\begin{array}{c} 2 \\ 1.0000000002 \\ 0.9999999998 \end{array}\right) & \begin{array}{c} x_1 \\ x_2 . \\ x_3 \end{array} \end{array}$$

Clearly, A is not ill-conditioned at all. But suppose we apply Gaussian elimination to solve the equation $A\underline{x} = \underline{b}$. Our first step could be to eliminate x_1 from equations 2 and 3 by subtracting suitable multiples of equation 1 from them. The reduced matrix should be

$$\begin{array}{cccc} \underline{x_1} & \underline{x_2} & \underline{x_3} & \\ \left(\begin{array}{c} 2 \times 10^{-10} \\ 0 \\ 0 \end{array}\right. & \begin{array}{c} -1 \\ -4999999999 \\ 5000000001 \end{array} & \left.\begin{array}{c} 1 \\ 5000000001 \\ -4999999999 \end{array}\right) & \begin{array}{c} b_1 \\ b'_2 \\ b'_3 \end{array} \end{array} .$$

But if we have a calculator whose capacity is limited to 8 decimal digits then the best we could do would be to approximate the reduced matrix by

$$\begin{array}{c} \begin{array}{ccc} \underline{x_1} & \underline{x_2} & \underline{x_3} \end{array} \\ \left(\begin{array}{ccc} 2 \times 10^{-10} & -1 & 1 \\ 0 & -5 \times 10^9 & 5 \times 10^9 \\ 0 & 5 \times 10^9 & -5 \times 10^9 \end{array}\right) \begin{array}{c} b_1 \\ b'_2 \\ b'_3 \end{array} ; \end{array}$$

but this is precisely the reduced matrix we should have obtained without rounding errors if A had originally been

$$A + \Delta A = \begin{array}{c} \begin{array}{ccc} \underline{x_1} & \underline{x_2} & \underline{x_3} \end{array} \\ \left(\begin{array}{ccc} 2 \times 10^{-10} & -1 & 1 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \end{array}\right) \begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array} . \end{array}$$

In other words, the data in A's lower right hand 2×2 submatrix has fallen off the right hand end of our calculator's 8-digit register, and been lost. The result is tantamount to distorting our original data by the amount of the loss, and in this example the distortion is a disaster.

These disasters occur whenever abnormally large numbers are added to the moderate sized numbers comprising our data. To avoid such disasters it is customary to choose the variable x_J to be eliminated, and/or the row I with which it is to be eliminated from all other rows, in such a way that the pivot a_{IJ} is the largest available element a_{ij} in its row, or column, or both. Since the typical computation replaces a_{ij} by

$$a'_{ij} = a_{ij} - a_{iJ}\, a_{Ij}/a_{IJ} \text{ for all } (i,j) \neq (I,J) ,$$

we see that $\max_{ij} |a'_{ij}| \leq 2 \max_{ij} |a_{ij}|$,

so that no abnormally large numbers should appear. In the example above we might choose a_{21} as the pivot to obtain the reduced matrix

$$\begin{array}{cccc}
\underline{x_1} & \underline{x_2} & \underline{x_3} & \\
\left(\begin{array}{c} -1 \\ 0 \\ 0 \end{array}\right. & \begin{array}{c} 1 \\ -1.0000000 \\ 2 \end{array} & \left.\begin{array}{c} 1 \\ 1.0000000 \\ 2 \end{array}\right) & \begin{array}{c} b_2 \\ b'_1 \\ b_3 \end{array}
\end{array}$$

(working to 8 significant digits) .

This reduced matrix is what would have resulted if no rounding errors had been committed during the reduction of

$$A + \Delta A = \begin{array}{cccc}
\underline{x_1} & \underline{x_2} & \underline{x_3} & \\
\left(\begin{array}{c} 0 \\ -1 \\ 1 \end{array}\right. & \begin{array}{c} -1 \\ 1 \\ 1 \end{array} & \left.\begin{array}{c} 1 \\ 1 \\ 1 \end{array}\right) & \begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array}
\end{array} ,$$

which differs negligibly from the given matrix A .

Wilkinson's error bound, quoted above, assumes that each pivot a_{IJ} is the largest in its row or else in its column of the reduced matrix, and then g_N is the ratio of the largest of the pivots to the largest element in A . We can see that, since the largest elements of each reduced matrix never exceed twice those of the previous one,

$$g_N \leq 2^{N-1} .$$

This bound is achieved for $\lambda = 1$ by the matrix

Example 3:

$$A_\lambda = \begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & & 0 & 1\\
-1 & 1 & 0 & 0 & \cdots & & 0 & 1\\
-1 & -1 & 1 & 0 & \cdots & & 0 & 1\\
-1 & -1 & -1 & 1 & & & \cdot & \cdot\\
\cdot & \cdot & \cdot & & \ddots & & \cdot & \cdot\\
\cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & & & & & \\
 & & & & & 1 & 0 & 1\\
 & & & & & -1 & 1 & 1\\
-1 & -1 & -1 & \cdots & & -1 & -1 & \lambda
\end{pmatrix}
\qquad
\begin{array}{l}
a_{ij} = -1 \text{ if } i > j ,\\
a_{ii} = \ \ 1 \text{ if } i < N ,\\
a_{ij} = 0 \text{ if } i < j < N,\\
a_{iN} = 1 \text{ if } i < N, \text{ and}\\
a_{NN} = \lambda .\\
\\
N \times N
\end{array}$$

if each pivot is chosen on the diagonal as one of the largest elements in its column, and the columns are chosen in their natural order $1,2,3, \ldots, N$ during the elimination. But when we repeat the computation with $\lambda = 2$ and sufficiently large N, an apparent disaster occurs because the value of $\lambda = 2$ gets lost off the right-hand side of our computing register. On a binary machine like our 7094 (using truncated 27-bit arithmetic), $\lambda = 2$ is replaced by $\lambda = 1$ if $N > 28$. An example like this was used by Wilkinson (1961, p. 327) as part of the justification for his recommendation that one use both row <u>and</u> column interchanges when selecting pivot a_{IJ} to ensure that it is one of the largest elements in the reduced matrix. This pivot-selection strategy is called "complete pivoting" to distinguish it from "partial pivoting" in which either row exchanges or column exchanges, but not both, are used.

The other justification for complete pivoting was Wilkinson's proof of a remarkable bound for the ratio g_N of the largest pivot to the largest element of A:

$$g_N \leq (N.2^1.3^{1/2}.4^{1/3} \ldots N^{1/(N-1)})^{1/2} \lesssim 2N^{\frac{1}{2}+\frac{1}{4}\log N} ,$$

which is certainly far smaller than 2^{N-1}.

(This bound is worth a small digression. It is known to be unachievable for $N > 2$; and the bound

$$g_N \leq N$$

has been conjectured for complete pivoting when A is real. The conjectured bound is achieved whenever A is a Hadamard matrix, and L. Tornheim has shown that the conjecture is valid when $N = 3$. He has also shown that when A is complex the larger bound

$$g_3 \leq 16/3^{3/2}$$

can be achieved.)

Despite the theoretical advantages of complete pivoting over partial pivoting, the former is used much less often than the latter, mainly because interchanging both rows and columns is far more of a nuisance than interchanging, say, rows alone. Moreover, it is easy to monitor the size of the pivots used during a partial pivoting computation, and stop the calculation if the pivots grow too large; then another program can be called in to recompute a more accurate solution with the aid of complete pivoting. Such is the strategy in use on our computer at Toronto, and the results of using this strategy support the conviction that intolerable pivot-growth is a phenomenon that happens only to numerical analysts who are looking for that phenomenon.

Despite the confidence with which the computed vector $\underline{z}$, produced by Gaussian elimination or some other comparable method, can be expected to have a residual

$$\underline{r} = \underline{b} - A\underline{z}$$

which is scarcely larger than the rounding errors committed during the calculation of $\underline{r}$,

$$\text{i.e. } \|\underline{r}\| = \|\underline{b} - A\underline{z}\| \sim N\beta^{-s}(\|\underline{b}\| + \|A\|\ \|\underline{z}\|) ,$$

an important problem remains. How large is the error $\underline{z} - \underline{x}$ with which $\underline{z}$ approximates the "true solution" $\underline{x}$ of $A\underline{x} = \underline{b}$? This question is meaningful even if A and $\underline{b}$ are not known precisely; we can interpret $\underline{z}$ to be the solution of a perturbed system

$$(A + \Delta A)\ \underline{z} = \underline{b} + \Delta\underline{b}$$

in which $\|\Delta A\|$ and $\|\Delta\underline{b}\|$ are bounded in some given way, and hence so is $\underline{r} = \Delta A\underline{z} - \Delta\underline{b}$. A precise answer to the question is

$$\underline{x} - \underline{z} = A^{-1}\underline{r} \ , \quad \|\underline{z} - \underline{x}\| \leq \|A^{-1}\|\ \|\underline{r}\| ,$$

but here we must know $\|A^{-1}\|$ in order to complete the answer. If we try to compute A^{-1} , we shall instead obtain some approximation, say Z , and once again we shall have to ask the question

How large is the error $Z - A^{-1}$?

This question can be answered fairly easily if Z is accurate enough, as shall now be shown.

Each column $\underline{z}_i$ of $Z = \{\underline{z}_1, \underline{z}_2, \ldots, \underline{z}_N\}$ can be regarded as an approximation to the corresponding column $\underline{x}_i$ of

$$A^{-1} = X = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\} ,$$

the solution of $AX = I$. A reasonable way to solve the last equation is to use Gaussian elimination, in which case each column $\underline{z}_i$ will be computed separately and will satisfy

$$(A + \Delta_i A)\ \underline{z}_i = (\text{the } i^{th} \text{ column of } I)$$

where $\|\Delta_i A\| \leq \varepsilon \|A\|$ for some small ε which depends upon the details of the program in a way discussed by Wilkinson (1961).

Now let R be the residual

$$R = I - AZ .$$

It is not necessary to compute R; we can write

$$R = I - A\ \{\underline{z}_1, \underline{z}_2, \ldots, \underline{z}_N\} = \{\Delta_1 A\underline{z}_1, \Delta_2 A\underline{z}_2, \ldots, \Delta_N A\underline{z}_N\},$$

in which each column $\Delta_i A\underline{z}_i$ of R is bounded in norm by

$$\|\Delta_i A\underline{z}_i\| \leq \varepsilon\|A\|\ \|\underline{z}_i\| .$$

Therefore

$$\rho = \|R\| \leq \eta\ \|A\|\ \|Z\|$$

where η/ε depends upon N and the norms; usually

$$\eta/\varepsilon \leq N^{1/2} .$$

Since ε can be predicted in advance, so can η; and it is possible to check whether

$$\eta\ \|A\|\ \|Z\| < 1 ,$$

in which case $\rho < 1$ and the following argument is valid:

$$\|A^{-1}\| \le \|Z\| + \|A^{-1}-Z\| = \|Z\| + \|A^{-1}(I-AZ)\| = \|Z\| + \|A^{-1}R\|$$
$$\le \|Z\| + \|A^{-1}\|\rho \;;$$
$$\|A^{-1}\| \le \|Z\|/(1-\rho) \;.$$

Then $\|A^{-1}-Z\| \le \|A^{-1}\|\rho \le \|Z\|\rho/(1-\rho)$.

The last inequality says that, neglecting modest factors which depend upon N , the relative error $\|A^{-1}-Z\|/\|Z\|$ is at worst about $K(A)$ times as large as the relative errors committed during each arithmetic operation of Z's computation. In other words, if A is known precisely then one must carry at least $\log_{10} K(A)$ more decimal guard digits during the computation of A^{-1} than one wants to have correct in the approximation Z , and one can verify the accuracy of Z at the end of the computation by computing $\eta\|A\|\|Z\|$.

If the method by which an approximation Z to A^{-1} was computed is not known, there is no way to check the accuracy of Z better than to compute R and ρ directly, and this calculation is not very attractive for two reasons. First, the computation of either residual

$$R = I-AZ \quad \text{or} \quad I-ZA$$

costs almost as much time as the computation of an approximate A^{-1} ; both computations cost about N^3 multiplications and additions. Second, if $K(A)$ is large then $\|I-AZ\|$ and $\|I-ZA\|$ can be very different, and there is no way to tell in advance which residual will give the least pessimistic over-estimate of the error in Z .

$$(\rho = \|I-AZ\| = \|A(I-ZA)A^{-1}\| \le K(A)\|I-ZA\| \text{ etc.})$$

Both residuals can be pessimistic by a factor like $K(A)$. Finally, although a better approximation to A^{-1} than Z is the matrix

$$Z_1 = Z + Z(I-AZ) = Z + (I-ZA)Z$$

(because $\|I-AZ_1\| = \|(I-AZ)^2\| \le \|I-AZ\|^2$) , the computation of Z_1 is in most cases more costly and less accurate than a direct computation of an approximate A^{-1} using Gaussian elimination with double precision arithmetic. For example, on our 7094 it takes less than twice as long to invert A to double precision (carrying 16 dec.) than to do the same job in single precision (8 dec.), and the double precision computation has almost 8 more correct digits in its answer. But Z_1 has at most twice as many correct digits as Z. Therefore, if Z comes from a single precision Gaussian elimination program, it will have about $8-\log K(A)$ correct digits. Z_1 will have $16-2\log K(A)$ digits at best. The double precision elimination will produce about $16-\log K(A)$ correct digits. Thus does engineering technique overtake mathematical ingenuity!

The solution of $A\underline{x} = \underline{b}$ for a single vector $\underline{x}$ is not normally performed by first computing A^{-1} and then $\underline{x} = A^{-1}\underline{b}$ for four reasons. First, the vector $Z\underline{b}$, where Z is an approximation to A^{-1}, is frequently much less accurate than the approximation $\underline{z}$ given directly by Gaussian elimination. Second, the direct computation of the vector $\underline{z}$ by elimination costs about 1/3 as much time as the computation of the matrix Z. Third, if one wants only to compute a vector $\underline{z}$ which makes $\underline{r} = \underline{b} - A\underline{z}$ negligible compared with the uncertainties in $\underline{b}$ and A, then Gaussian elimination is a satisfactory way to do the job despite the possible ill-condition of A, whereas $\underline{b} - A(Z\underline{b}) = R\underline{b}$ can be appreciably larger than negligible. Fourth, Gaussian elimination can be applied when A is a band matrix without requiring the vast storage that would otherwise be needed for A^{-1}. 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 error-bound for $\underline{z} - \underline{x}$. This disadvantage can be partially overcome by an iterative method known as re-substitution.

To solve $A\underline{x} = \underline{b}$ by re-substitution, we first apply any direct method, say Gaussian elimination, to obtain an approximation $\underline{z}$ to $\underline{x}$.

This vector $\underline{z}$ will be in error by $\underline{e} = \underline{x} - \underline{z}$, and

$$A\underline{e} = A(\underline{x}-\underline{z}) = \underline{b} - A\underline{z} = \underline{r} ,$$

which can be computed. (It is necessary to compute $\underline{r}$ carefully lest it consist of nothing but the rounding errors left when $\underline{b}$ and $A\underline{z}$ nearly cancel. Double precision accumulation of products is appropriate here.) Clearly, the error $\underline{e}$ satisfies an equation similar to $\underline{x}$'s except that $\underline{r}$ replaces $\underline{b}$. Therefore, we can approximate $\underline{e}$ by $\underline{f}$, say, obtained by repeating part of the previous calculation. If enough intermediate results have been saved during the computation of $\underline{z}$, one obtains $\underline{f}$ by repeating upon $\underline{r}$ the operations that transformed $\underline{b}$ into $\underline{z}$. The cost of $\underline{f}$ in time and storage is usually negligible.

Now $\underline{z}' = \underline{z} + \underline{f}$ is a better approximation to $\underline{x}$ than was $\underline{z}$, provided $\underline{z}$ was good enough to begin with. We shall see that this is so in the case of Gaussian elimination as follows:

$$(A + \Delta_1 A)\, \underline{z} = \underline{b} \text{ , where } \|\Delta_1 A\| \le \varepsilon \|A\| .$$

$$\underline{r} = \underline{b} - A\underline{z} \text{ , say exactly for simplicity.}$$

$$\|\underline{r}\| = \|\Delta_1 A\underline{z}\| \le \varepsilon \|A\|\ \|\underline{z}\| ,$$

and this inequality is not normally a wild overestimate.

$$(A + \Delta_2 A)\, \underline{f} = \underline{r} , \quad \|\Delta_2 A\| \le \varepsilon \|A\| .$$

$$\underline{r}' = \underline{b} - A\underline{z}' = \underline{r} - A\underline{f} = \Delta_2 A\underline{f} \text{ , so}$$

$$\|\underline{r}'\| \le \varepsilon \|A\|\ \|\underline{f}\| \le \varepsilon \|A\|\ \|(A + \Delta_2 A)^{-1}\|\ \|\underline{r}\|$$

$$\le \varepsilon\ K(A)\ \|\underline{r}\|/(1 - \varepsilon\ K(A)) \text{ if } \varepsilon\ K(A) < 1 .$$

And if $\varepsilon\ K(A) << 1$ then $\|\underline{r}'\|$ can be expected to be much smaller than $\|\underline{r}\|$. If $\underline{z}'$ is renamed $\underline{z}$, the process can be continued. We have left out several details here; the point is that the process of re-substitution generally converges to an approximation $\underline{z}$ which is correct to nearly full single precision, provided the matrix A is farther from a singular matrix than a few hundred units in its last place. The problem is to know when to stop.

The word "convergence" is well-defined mathematically in several contexts. But the empirical meaning of "convergence" is more subtle. For example, suppose we consider the sequence $\underline{z}_1, \underline{z}_2, \ldots, \underline{z}_n, \ldots$ of successive approximations to $\underline{x}$ produced by the re-substitution iteration, and suppose that $\underline{z}_m = \underline{z}_{m+1} = \underline{z}_{m+2} = \ldots$. We should conclude that the sequence has converged. And if $\underline{z}_{m-1} - \underline{z}_m$ is a good deal smaller than $\underline{z}_{m-2} - \underline{z}_{m-1}$, we should incline to the belief that the convergence of the sequence is not accidental; there is every reason to expect $\underline{z}_m$ to be the correct answer $\underline{x}$ except for roundoff in the last place. But a surprise is possible if A is exceptionally ill-conditioned:

Example 4. Here is an example of a 2x2 system with

$$A = \begin{pmatrix} .8647 & .5766 \\ .4322 & .2882 \end{pmatrix} \quad \text{and} \quad \underline{b} = \begin{pmatrix} .2885 \\ .1442 \end{pmatrix} .$$

We shall use Gaussian elimination to compute a first approximation $\underline{z}$ to the solution $\underline{x}$ of $A\underline{x} = \underline{b}$. Then $\underline{r} = \underline{b} - A\underline{z}$ is computed exactly, and the solution $\underline{e}$ of $A\underline{e} = \underline{r}$ is approximated by $\underline{f}$, obtained again by Gaussian elimination. $\underline{z}' = \underline{z} + \underline{f}$, and $\underline{r}' = \underline{b} - A\underline{z}'$.

We shall try to calculate $\underline{x}$ correctly to 3 sig. fig's. It seems reasonable to carry one guard digit at first, since we can repeat the calculation with more figures later if that is not enough. We shall truncate all calculations to 4 sig. fig's., just like our 7094 (except that it truncates to about 8 sig. fig's.). Intermediate results enclosed in parentheses are obtained by definition rather than by means of an arithmetic operation.

Comment	Equ'n no.	Coef. of x_1	Coef. of x_2	Right hand side $\underline{b}$
1st pivotal row is ...	E1	.8647	.5766	.2885
.4327/.8647=.4998	E2	.4322	.2882	.1442
.4998xE1 is ...	E1'	(.4322)	.2881	.1441
E2-E1' ...	E3	(0)	$.1x10^{-3}$	$.1x10^{-3}$
E3/$.1x10^{-3}$ ...	Z2	(0)	(1)	1.000
.5766xZ2 ...	E3'	(0)	(.5766)	.5766
E1-E3' ...	E4	(.8647)	(0)	-.2881
E4/.8647 ...	Z1	(1)	(0)	-.3331

Our first approximation is $\underline{z} = \begin{pmatrix} -.3331 \\ 1.000 \end{pmatrix}$.

Next we compute $\underline{r} = \underline{b} - A\underline{z}$ exactly using double precision:

Residual of E1 ...	R1	$-.6843x10^{-4}$
Residual of E2 ...	R2	$-.3418x10^{-4}$

We get $\underline{f}$ by repeating upon $\underline{r}$ the operation which transformed $\underline{b}$ into $\underline{z}$.

.4998xR1 ...	R1'	$-.3420x10^{-4}$
R2-R1' ...	R3	$.2x10^{-7}$
R3/$.1x10^{-3}$ ...	F2	$.2000x10^{-3}$
.5766xF2 ...	R3'	$.1153x10^{-3}$
R1-R3' ...	R4	$-.1837x10^{-3}$
R4/.8647 ...	F1	$-.2124x10^{-3}$

$$\underline{z} = \begin{pmatrix} .3331 \\ 1.000 \end{pmatrix} , \ \underline{f} = \begin{pmatrix} -.2124 \\ .2000 \end{pmatrix} x10^{-3} , \ \underline{z}'=\underline{z}+\underline{f}= \begin{pmatrix} -.3333124 \\ 1.0002000 \end{pmatrix} .$$

Clearly $\underline{z}'$ is so close to $\underline{z}$ that either is an acceptable 3-figure approximation to $\underline{x}$. But, just in case there is some doubt, we compute

$$\underline{r}' = \underline{b} - A\underline{z}' = \begin{pmatrix} -.00000008772 \\ .00000002072 \end{pmatrix} ,$$

which is reassuringly smaller than $\underline{r}$.

Is $\underline{x} = \begin{pmatrix} -.333 \\ 1.000 \end{pmatrix}$ to 3 sig. fig's? No, $\underline{x} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$.

The only clue to A's ill-condition is the cancellation in E3 . If there were time and space available, an example could be constructed of sufficiently high dimensionality that severe cancellation would not occur to warn of the disaster. Presumably this kind of disaster is rare in practice, because none has yet been reported elsewhere. Indeed, a prominent figure in the world of error-analysis has said

> "Anyone unlucky enough to encounter this sort of calamity has probably already been run over by a truck."

But being run over by a truck can hardly go unnoticed.

Despite the risks, re-substitution is the most reasonable and efficient way to check and improve the accuracy of an approximation $\underline{z}$ when the matrices A and $\underline{b}$ are known more precisely than to within the uncertainties ΔA and $\Delta\underline{b}$ in the perturbed equation

$$(A + \Delta A)\,\underline{z} = \underline{b} + \Delta\underline{b}$$

satisfied by the product $\underline{z}$ of Gaussian elimination. For fuller detail, see Wilkinson (1963) pp. 121-126. But if A and $\underline{b}$ are intrinsically uncertain in, say, their fourth decimal place, and if Gaussian Elimination has been carried out with about 6 to 8 sig. fig's and with a reasonable pivotal strategy, then re-substitution may well be pointless, since the errors committed during the elimination will be negligible compared with intrinsic uncertainties.

5. Pivoting and Equilibration. How reliable are the sizes of the pivots as indications of a matrix's ill-condition? Is it true that a matrix is ill-conditioned if and only if some of its pivots are small? Most people who are experienced with hand calculations would answer "yes" to the last question unless they have tried to test their belief on problems of high dimensionality with the aid of an electronic computer. When the dimensionality of a

problem becomes large (say > 30), much of our experience and intuition with small dimensionality (say ≤ 5) becomes misleading. The following examples are designed to correct misleading impressions.

MIS-STATEMENT NO. 1. The determinant of A is the product of the pivots encountered during Gaussian elimination (to within a ± sign); and since a singular matrix has determinant zero, and $\det A$ is a continuous function of A, an ill-conditioned (nearly singular) matrix must have a small determinant and hence must have at least one small pivot.

The flaw in this argument is the same as that which says that, since $|x|^{1/N}$ is a continuous function of x no matter how large N may be, $|x|^{1/N}$ must be small when x is small. The trouble is that two "small" numbers can still be relatively very different. A matrix counterexample is

Example 5.

$$A = \begin{pmatrix} 1 & -1 & -1 & \cdots & -1 \\ 0 & 1 & -1 & \cdots & -1 \\ 0 & 0 & 1 & \cdots & -1 \\ \vdots & & & \ddots & \vdots \\ & & & 1 & -1 \\ 0 & \cdots & & & 1 \end{pmatrix}_{N\mathrm{x}N} \quad \begin{aligned} a_{ij} &= 0 \text{ if } i > j \,, \\ a_{ij} &= -1 \text{ if } i < j \,, \\ a_{ii} &= 1 \,. \end{aligned}$$

Here $\det A = 1$, and every pivot can be 1, but A can be made singular by subtracting 2^{1-N} from all a_{i1}. Therefore, when N is large A differs negligibly from a singular matrix, and must be ill-conditioned. The ill-condition of A is not "caused" by a large number of nearly equal elements, as some observers have suggested, because |if all -1's in A are replaced by $+1$'s then A becomes the well conditioned inverse of

$$\begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ 0 & 0 & 1 & . & 0 \\ \vdots & & & \ddots & \vdots \\ & & & 1 & -1 \\ 0 & \cdots & & 0 & 1 \end{pmatrix} .$$

The foregoing examples indicate that Gaussian elimination is a poor way to determine the rank of a matrix because a few rounding errors may suffice to cause none of the pivots to be small despite a theorem which says that, in the absence of rounding errors, the rank of a matrix is the same as the number of non-zero pivots generated during Gaussian elimination with both row and column pivotal interchanges to select maximal pivots.

Most other methods for determining rank fare no better in the face of roundoff. For example, the Schmidt orthogonalization procedure can be described in terms of an orthogonal projection of A's n-th column $\underline{a}_n$ upon the space spanned by the previous n-1 columns $\underline{a}_1, \ldots, \underline{a}_{n-1}$ (see Householder (1964) pp. 6-8 and 134-7). The columns can be interchanged, if necessary, in order at each stage to maximize the distance from $\underline{a}_n$ to the space of $\underline{a}_1, \ldots, \underline{a}_{n-1}$. If this is done, the rank r of A will become evident when $\underline{a}_{r+1}, \underline{a}_{r+2}, \ldots,$ and $\underline{a}_N$ all have distance zero from the space spanned by $\underline{a}_1, \underline{a}_2, \ldots,$ and $\underline{a}_r$. However, if A is merely nearly singular, there is no guarantee that any of the distances mentioned above will be anywhere near as small as the distance between A and the nearest singular matrix. Difficulties arise whenever $\|A\underline{v}\| << \|A\| \; \|\underline{v}\|$ for some vector $\underline{v}$ whose components v_i can be ordered in such a way that they steadily decrease in magnitude to a point where the smallest component is negligible compared with the largest. The following example illustrates the phenomenon:

$$\left.\begin{array}{ll} a_{ij} = 0 & \text{if} \quad i > j \\ a_{ii} = s^{i-1} & \\ a_{ij} = -cs^{i-1} & \text{if} \quad i < j \end{array}\right\} \quad \text{for } i,j = 1,2, \ldots , N$$

Here N is large $(N > 30)$ and $s^2 + c^2 = 1$. Since A is upper triangular, the n^{th} column of A is distant a_{nn} from the space spanned by the previous columns. Also, this example has been so chosen that no column interchanges are needed to maximize the distance, since $\underline{a}_n, \underline{a}_{n+1}, \ldots$, and $\underline{a}_N$ are all equally distant from the space spanned by $\underline{a}_1$, $\underline{a}_2$, ... , $\underline{a}_{n-1}$. The smallest distance a_{nn} is $a_{NN} = s^{N-1}$. How much smaller than a_{NN} is the distance $\|\Delta A\|$ between A and the nearest singular matrix $A + \Delta A$?

By examining the vector $A\underline{v}$, where

$$v_i = c(1+c)^{N-i-1} \quad \text{except} \quad v_N = 1 ,$$

we can show that

$$\|\Delta A\|/a_{NN} = O(1/(1+c)^{N-1}) \quad \text{as} \quad N \to \infty .$$
$$= O(1/(1+\sqrt{1-a_{NN}{}^{2/N}})^N) .$$

For fixed $a_{NN} = \alpha > 0$, the righthand side tends to zero like

$$1/[\alpha \exp\sqrt{-2\ (N - 1/3 \log \alpha) \log \alpha}\ ] \quad \text{as } N \to \infty .$$

For fixed N , it is like 2^{-N} as $a_{NN} \to 0$. In other words, A can be closer to singular than a_{NN} by orders of magnitude if N is large.

No simple method is known for computing the rank of a matrix in the face of roundoff. An effective but complicated method has been given by Golub and Kahan (1965).

MIS-STATEMENT NO. 2. The reason for pivotal interchanges is to prevent incorrect answers caused by the use of an inaccurate small pivot.

This statement seems reasonable in the light of the 2x2 example at the end of section 4 of this paper, where cancellation in E3 produced a tiny pivot .0001 whose value consisted almost entirely of rounding error. And the computed answer was quite wrong. Post hoc ergo propter hoc. However, the pivot $2\text{x}10^{-10}$ in example 2 is quite accurate, yet any answer gained by its use is likely to be wrong.

Now, what can one mean by the accuracy of a pivot? The meaning would be clear if the object of our computation were to calculate pivots, but that is not our object. We wish to satisfy a set of linear equations, and the computed solution $\underline{z}$ can come very near to satisfying $A\underline{x} = \underline{b}$ even though almost all pivots are entirely different from what they would have been in the absence of roundoff. In example 4, R1 and R2 are as small as one could reasonably expect from 4-figure working, and remain so even when the pivot .0001 is replaced by, say, .0002 . There are occasions when a small residual is all that is wanted (see the discussion of eigenvector calculations below). In such a case, we must conclude that small pivots have not prevented a correct answer from being produced. Besides, pivotal interchanges do not prevent small pivots.

What is the significance of a small pivot? In the absence of other information, none. For example, if A is a diagonal matrix

$$a_{ii} = 10^{-i} \quad \text{exactly} \ ,$$

then the system $A\underline{x} = \underline{b}$ can be solved trivially and precisely (in decimal arithmetic) despite tiny pivots. On the other hand, if we are given a matrix norm such that all perturbations ΔA of equal norm $\|\Delta A\|$ are considered equally important, and if we measure the ill-condition of A in terms of this norm, then a small pivot tells us that A is ill-conditioned as follows:

If the rows and columns of A are ordered properly to begin with, the process of Gaussian elimination can be identified with a triangular factorization

$$A = LU - E$$

where L is unit lower triangular($\ell_{ij}=0$ if $i<j$ and $\ell_{ii}=1$),
U is upper triangular ($u_{ij}=0$ if $i>j$) , and
E is the contribution of roundoff.
(See Wilkinson (1963).)
If partial pivoting is used, $|\ell_{ij}| \leq 1$ for $i \geq j$.
If complete pivoting is used, $|u_{ij}| \leq |u_{ii}|$ too.
The pivots are the numbers u_{ii} , and

$$\|E\| \leq \varepsilon \|L\| \, \|U\|$$

where ε is comparable with the relative error associated with one rounding error (ε is about 10^{-8} on our machine).

Suppose, now, that some pivot u_{ii} is small. Then let $U+\Delta U$ differ from U only in that u_{ii} is replaced by zero. Therefore,

$$A + \Delta A = A + E + L\Delta U = L(U + \Delta U) \text{ is singular .}$$

And $\|\Delta A\|$ is of the order of $\|L\|(\varepsilon\|U\| + |u_{ii}|)$.

Since $1 \leq \|L\| < N$ for most norms of interest here, and $\|U\| < Ng_N \|A\|$ in most cases of interest (g_N was the pivot-growth ratio), one perturbation ΔA that makes $A+\Delta A$ singular is of the same order of magnitude as the smallest pivot u_{ii} . And since $\|\Delta A\| \geq \|A\|/K(A)$, the condition number $K(A) > \|A\|/(N|u_{ii}|)$. A small pivot implies ill-condition with respect to the given norm.

The foregoing argument also shows why pivotal interchanges are necessary. They help to keep $\|U\|$ from growing too large, thereby contributing to keeping $\|E\|$ small, and this last is what we want. The error in example 2 when $2\text{x}10^{-10}$ is used as a pivot illustrates the consequences of allowing $\|U\|$ to grow too large. Had any other element of A been chosen as a pivot, no such error could have occurred.

MIS-STATEMENT NO. 3. If A's condition number $K(A)$ is very large, and if A and $\underline{b}$ are uncertain by a few units in their last place, then no numerical method is capable of solving $A\underline{x} = \underline{b}$ more accurately than to about $K(A)$ units in $\underline{x}$'s last place.

This statement would be true if the uncertainties in A , $\underline{b}$ and $\underline{x}$ were measured in the norms corresponding to $K(A)$. The most appropriate norms for $\underline{b}$ and $\underline{x}$ would be such that perturbations of equal norm were equally likely or equally costly or otherwise practically indistinguishable. But rarely in practice is an appropriate norm chosen on that basis. Usually one of the Hölder norms is chosen on the basis of convenience, and such norms can be terribly inappropriate.

Example 6.

$$\text{Let } \bar{A} = \begin{pmatrix} 2 & -1 & 1 \\ -1 & 10^{-10} & 10^{-10} \\ 1 & 10^{-10} & 10^{-10} \end{pmatrix} \text{ and } \underline{b} = \begin{pmatrix} 2 \\ -10^{-10} \\ 10^{-10} \end{pmatrix},$$

with a relative uncertainty of 10^{-8} in each element. In other words, $\bar{A} + \Delta\bar{A}$ is acceptable in place of $\bar{A}$ provided $|\Delta\bar{a}_{ij}| \le 10^{-8} |\bar{a}_{ij}|$. If any of the aforementioned Hölder norms are used, the condition number of $\bar{A}$ is at least 10^{10} because there exists a $\Delta\bar{A}$ with $\|\Delta\bar{A}\| \le 10^{-10} \|\bar{A}\|$ such that $\bar{A}+\Delta\bar{A}$ is singular. Therefore, when Gaussian elimination carried out with eight sig. fig. arithmetic gives no useful answer, one is not surprised. "The system is ill-conditioned." However, the true solution is

$$\underline{x} = \begin{pmatrix} 10^{-10} \\ -1 \\ 1 \end{pmatrix}$$

with a relative error smaller than 10^{-7} in each component no matter how $\bar{A}$ and $\underline{b}$ are perturbed, provided only that no element of $\bar{A}$ or $\underline{b}$ is changed by more than 10^{-8} of itself. This system is well conditioned! But not in the usual Hölder norm.

Example 6 can be obtained from example 2 by a diagonal transformation;

$$\bar{A} = DAD \quad \text{with} \quad D = \text{diag } (10^{5} , 10^{-5} , 10^{-5}) .$$

This transformation does no more than shift a decimal point 10 places left or right. If Gaussian elimination is applied simultaneously to the matrices A and $\bar{A}$, then the results in both cases will be identical down to the rounding errors except for the 10 place shifts of decimal point. But example 2 teaches us not to use 2×10^{-10} as a pivot, whereas the most natural pivot in example 6 is the corresponding element 2. Any other pivot would be far better.

This is where equilibration comes in. Equilibration consists of diagonal transformations intended to scale each row and column of A in such a way that, when Gaussian elimination is applied to the equilibrated system of equations, the results are nearly as accurate as possible. In other words, the system

$$A\underline{x} = \underline{b}$$

is replaced by

$$(RAC)\ \underline{y} = (R\underline{b})$$

where R and C are diagonal matrices. Then Gaussian elimination (or any other method) is applied to the array $\{RAC, R\underline{b}\}$ to produce an approximation to $\underline{y}$ and hence to $\underline{x} = C\underline{y}$.

How should R and C be chosen? No one has published a foolproof method. The closest anyone has come is in a paper by F.L. Bauer (1963) in which the R and C which minimize $K(RAC)$ are described in terms of an eigenvalue and eigenvectors of certain matrices constructed from A and A^{-1} . But no way is known to construct R and C without first knowing A^{-1} .

There is some doubt whether R and C should be chosen to minimize $K(RAC)$. The next example illustrates the problem; the reader should write out the matrices involved in extenso for $N = 6$ to follow the argument.

Let A_λ be the NxN matrix defined in example 3, and let

$$R = \mathrm{diag}\ (\tfrac{1}{2}\ ,\ \tfrac{1}{4}\ ,\ \ldots\ ,\ 2^{2-N}\ ,\ 2^{1-N}\ ,\ 2^{1-N})\ \text{and}$$

$$C = \mathrm{diag}\ (1\ ,\ 2\ ,\ \ldots\ ,\ 2^{N-3}\ ,\ 2^{N-2}\ ,\ 1)\ .$$

We observe that $A_1{}^{-1} = (RA_1C)^\tau$, so A_1 and $A_1{}^{-1}$ are both well-conditioned matrices with elements no larger than 1.

Let $\underline{u} = (0, 0, 0, \dots, 0, 1)^\tau$ and note that

$$A_2 = A_1 + \underline{u}\,\underline{u}^\tau .$$

Therefore, by a simple computation,

$$A_2^{-1} = A_1^{-1} - (1 + 2^{1-N})^{-1} A_1^{-1}\,\underline{u}\,\underline{u}^\tau A_1^{-1} .$$

Now recall that when row-pivoting alone is used with A_2, the result can be an error tantamount to replacing A_2 by A_1, and if no other errors are committed then one will compute A_1^{-1} instead of A_2^{-1}. Note that, when the usual Hölder norms are used,

$$\|A_1-A_2\|/\|A_1\| = O(1) \quad \text{and} \quad \|A_1^{-1}-A_2^{-1}\| = O(1) .$$

Next observe that either complete or partial pivoting can be used with RA_2C ; the result is to make an error which replaces RA_2C by RA_1C. But now

$$\|RA_1C - RA_2C\|/\|RA_1C\| = O(2^{-N}) \quad \text{and}$$

$$\|(RA_1C)^{-1} - (RA_2C)^{-1}\|/\|(RA_2C)^{-1}\| = O\,(2^{-N}) .$$

In other words, despite some hocus-pocus with shifted binary points, the error made in applying Gaussian elimination to RA_2C is the same as that made with A_2, except for a change of scale. But the former error looks negligible and affects $(RA_2C)^{-1}$ negligibly, whereas the same errors look disastrous in A_2 and A_2^{-1}. And nowhere is there any ill-conditioned matrix, nor do any of the matrices look poorly scaled by the usual criteria!

The moral of the story is that the choice of R and C should reflect the norms by which the errors are being appraised. But no one knows yet precisely how to effect such a choice.

Though I do know roughly.

REFERENCES

1. D.W. Barron and H.P.F. Swinnerton-Dyer, Solution of Simultaneous Linear Equations Using a Magnetic-Tape Store. The Computer Journal, Vol. 3, (1960) pp. 28-33.

2. F.L. Bauer, Optimally Scaled Matrices. Numerische Math., Vol. 5, (1963) pp. 73-87.

3. G. Birkhoff, R.S. Varga, and D. Young, Alternating Direction Implicit Methods. Advances in Computers, Vol. 3, Academic Press (1962).

4. E. Bodewig, Matrix Calculus, 2nd ed. North Holland (1959). (A catalogue of methods and tricks, with historical asides.)

5. M.A. Cayless, Solution of Systems of Ordinary and Partial Differential Equations by Quasi-Diagonal Matrices. The Computer Journal, Vol. 4, (1961) pp. 54-61.

6. M.M. Day, Normed Linear Spaces. Springer (1962).

7. J. Douglas and J.E. Gunn, A General Formulation of Alternating Direction Methods, part I. Numerische Math. Vol. 6, (1965) pp. 428-453.

8. Dunford and Schwartz, Linear Operators, part I: General Theory. Interscience (1958).

9. M. Engeli et. al., Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems. Mitteilung Nr. 8 aus dem Inst. für angew. Math. an dur E.T.H., Zurich: Birkhauser (1959).

10. D.K. Faddeev and V.N. Faddeeva, Computational Methods of Linear Algebra, translated from the Russian by R.C. Williams. W.H. Freeman (1964). (This text is a useful catalogue, but weak on error-analysis. A new augmented Russian edition has appeared.)

11. G.E. Forsythe and W.R. Wasow, Finite Difference Methods for Partial Differential Equations. Wiley (1960). (A detailed text.)

12. L. Fox, The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations. Oxford Univ. Press (1957).

13. L. Fox, Numerical Solution of Ordinary and Partial Differential Equations. Pergamon Press (1962) (Ed.). (Based on a Summer School held in Oxford, Aug.-Sept. 1961.)

14. L. Fox, An Introduction to Numerical Linear Algebra. Oxford Univ. Press (1964). (This is an excellent introduction.)

15. C.F. Gauss, Letter to Gerling, 26 Dec. 1823. Werke Vol. 9, (1823) pp. 278-281. A translation by G.E. Forsythe appears in MTAC Vol. 5 (1950), pp. 255-258.

16. C.F. Gauss, Supplementum ... . Werke, Göttingen, Vol. 4, (1826) pp. 55-93.

17. G. Golub and W. Kahan, Calculating the Singular Values and Pseudo-Inverse of a Matrix. J. SIAM Numer. Anal. (B), Vol. 2, (1965) pp. 205-224.

18. J.E. Gunn, The Solution of Elliptic Difference Equations by Semi-Explicit Iterative Techniques. J. SIAM Numer. Anal. Ser. B, Vol. 2, (1965) pp. 24-45.

19. A.S. Householder, The Theory of Matrices in Numerical Analysis. Blaisdell (1964). (An elegant but terse treatment, including material on matrix norms which is otherwise hard to find in Numerical Analysis texts.)

20. IFIP: Proceedings of the Congress of the International Federation for Information Processing, held in New York City, May 24-29, 1965. Spartan Books (1965).

21. C.G.J. Jacobi, Über eine neue Auflösumgsart ... . Astr. Nachr. Vol. 22, No. 523, (1845) pp. 297-306. (Reprinted in his Werke Vol. 3, p. 467.)

22. L.V. Kantorovich and G.P. Akilov, Functional Analysis in Normed Spaces, translated from the Russian by D.E. Brown. Pergamon (1964).

23. R.B. Kellog and J. Spanier, On Optimal Alternating Direction Parameters for Singular Matrices. Math. of Comp., Vol. 19, (1965) pp. 448-451.

24. V.V. Klyuyev and N.I. Kokovkin-Shcherbak, On the Minimization of the Number of Arithmetic Operations for the Solution of Linear Algebraic Systems of Equations. Journal of Computational Math. and Math. Phys., Vol. 5, (1965) pp. 21-33 (Russian). A translation, by G.J. Tee, is available as Tech. Rep't CS24 from the Computer Sci. Dep't of Stanford University. (My copy has mistakes in it which I have not yet sorted out.)

25. J. Liouville, Sur le dévelopement des fonctions en series ... . II, J. Math. pures appl. (1), Vol. 2, (1837) pp. 16-37.

26. D.W. Martin and G.J. Tee, Iterative Methods for Linear Equations with Symmetric Positive Definite Matrix. The Computer Journal Vol. 4, (1961) pp. 242-254. (An excellent survey.)

27. W.A. Murray and M.S. Lynn, A Computer-Oriented Description of the Peaceman-Rachford ADI Method. The Computer Journal, Vol. 8, (1965) pp. 166-175.

28. J. von Neumann and H.H. Goldstine, Numerical Inverting of Matrices of High Order. Bull. Amer. Math. Soc., Vol. 53, (1947) pp. 1021-1099.

J. von Neumann and H.H. Goldstine, ".... part II". Proc. Amer. Math. Soc., Vol. 2, (1951) pp. 188-202.

29. L.B. Rall, Error in Digital Computation. Two volumes (1965) Wiley. (Contains a valuable bibliography.)

30. G.D. Smith, Numerical Solution of Partial Differential Equations. Oxford Univ. Press (1965). (This is an introductory text.)

31. E.L. Stiefel, Some Special Methods of Relaxation Technique appearing in Simultaneous Linear Equations and the Determination of Eigenvalues. National Bureau of Standards Applied Math. Series No. 29 (1953). (A subsequent article by J.B. Rosser in this same book contains more details about conjugate gradient methods.)

32. E.L. Stiefel, Kernel Polynomials in Linear Algebra and their Applications. in Further Contributions ..., National Bureau of Standards Applied Math., Series No. 49 (1958).

33. A.M. Turing, Rounding-off Errors in Matrix Processes. Quart. J. Mech. Appl. Math. 1, (1948) pp. 287-308.

34. R.S. Varga, Matrix Iterative Analysis. Prentice Hall (1962). (An important treatise on those iterative methods most widely used to solve large boundary-value problems.)

35. J.H. Wilkinson, Rounding Errors in Algebraic Processes. in Information Processing, (1960) pp. 44-53. Proceedings of a UNESCO conference held in Paris in 1959.

36. J.H. Wilkinson, Error Analysis of Direct Methods of Matrix Inversion. J. Assoc. Computing Machinery, Vol. 8 (1961) pp. 281-330.

37. J.H. Wilkinson, Rounding Errors in Algebraic Processes. National Physical Lab. Note on Applied Science No. 32 (1963), Her Majesty's Stationery Office.

University of Toronto