An empirical estimate of the relative error of the computed solution $\bar{x}$ of $Ax = b$

By G. Loizou*

An empirical method is described for providing, with a minimum of effort, useful estimates for the relative errors in individual components of the computed solution $\bar{x}$ of the system of linear equations $Ax = b$. The estimate for the error is in no sense a rigorous upper bound, but in view of the economy of effort it is felt that extensions of the method described here provide a valuable addition to known methods of *a posteriori* error analyses.

(First received June 1967, and in revised form, September 1967)

1. Introduction

The problem to be discussed in this paper is that of estimating empirically the accuracy of the computed vector $\bar{x}$, obtained in the solution of

$$Ax = b, \tag{1}$$

where A will be assumed to be an $n \times n$ non-singular matrix. The computed solution can be effected by either a direct or an iterative method. Here attention is given to the direct method of Gaussian elimination which has been studied extensively by Wilkinson (1963 and 1965). As is often the case the matrix A is very or fairly ill-conditioned and the computed solution $\bar{x}$ is a poor approximation to the true solution. The reason for this is attributed to the accumulation of rounding errors, cancellation and the ill-conditioning of the matrix A. The accumulation of rounding errors has been studied (see, for example, Wilkinson, 1963 and 1965) by using the elegant technique of backward error analysis in floating or fixed-point arithmetic. However, once a computed solution $\bar{x}$ is obtained one would like to know approximately the accuracy. In this paper a practical method is developed which gives component-wise estimates of the accuracy of the computed solution $\bar{x}$. It must be emphasised that the experimental relative error obtained by the empirical method described in this paper is not an upper bound of the actual relative error in the computed solution; it does, however, give a good approximation to the order of magnitude of the actual relative error. The method, as it will be seen, was applied to very ill-conditioned matrices and the results have been fairly good. Tables are given for various matrices and, as is shown from the ALGOL procedure of the method, almost no extra computing time is required to ascertain the accuracy of $\bar{x}$, since most of the quantities needed to estimate it are obtained progressively in the various stages of the elimination method for solving (1).

2. Description of the method

If A is a non-singular matrix then, in general, it can be factorised in the form $A = LU$, where L is unit lower triangular and U is upper triangular (Gaussian elimination). The factorisation, when it exists, is unique.

Denote the initial set of equations by $A_0 x = b^0$. The factorisation which is performed in $n - 1$ major steps is equivalent to premultiplication of A_0 by elementary matrices M_i, $i = 1(1)n - 1$, where the ith step is given by

$$A_i = M_i M_{i-1} \ldots M_2 M_1 A_0 = P_i A_0,$$
$$i = 1(1)n - 1,\ P_{n-1} = L^{-1} \text{ and } A_{n-1} = U.$$

The forward substitution, which gives $Ux = L^{-1}b^0$, is similarly equivalent at the ith step to

$$b^i = M_i M_{i-1} \ldots M_2 M_1 b^0 = P_i b^0,$$
$$i = 1(1)n - 1, \text{ and } b^{n-1} = L^{-1}b^0.$$

The backward substitution gives progressively the computed solution vector $\bar{x}$, whose components $\bar{x}_r$, $r = n(-1)1$, are given by

$$\bar{x}_r = \frac{u.y}{u_{rr}},\ r = n(-1)1, \quad \text{where}$$
$$u = (-u_{r\,r+1}, -u_{r\,r+2}, \ldots, -u_{rn}, b_r^{n-1})^T,$$
$$y = (\bar{x}_{r+1}, \bar{x}_{r+2}, \ldots, \bar{x}_n, 1)^T,$$

and u^T denotes the transpose of u.

Having considered the above structure of the solution of (1), an empirical method is now devised for computing the component-wise relative loss of accuracy of the computed solution vector x. Define

$$p_r = \frac{|u_{rr}|}{\max\limits_{i=0(1)r-1} |a_{rr}^{(i)}|}, \qquad r = 1(1)n,$$

$$q_r = \frac{||(u_{r\,r+1}, u_{r\,r+2}, \ldots, u_{rn}, L_r^{-1}b^0)||}{\max\limits_{i=0(1)r-1} ||(a_{r\,r+1}^{(i)}, a_{r\,r+2}^{(i)}, \ldots, a_{rn}^{(i)}, b_r^i)||}, \quad r = 1(1)n,$$

where $a_{n\,n+1}^{(n-1)} = u_{n\,n+1} = 0$, $L_r^{-1}.$ denotes the r throw of the matrix L^{-1}, the underlying norm $||.||$ being the euclidean vector norm, and

$$s_r = \frac{|u.y|}{||u||\,||y||}, \quad r = 1(1)n.$$

REMARK. The quantities p_r, q_r estimate respectively the loss of relative accuracy in the rth pivot and the remaining part of the rth pivotal row.

* *Computer Centre, Queen Mary College, London, E.*1.
*Present address: Mathematics Department, Birkbeck College, London, W.C.*1.

A quantity t_r is defined progressively (as the elements of $\bar{x}, \bar{x}_r, r = n(-1)1$, are computed in the backward substitution) in the following way:

$$t_r = \frac{[1 + (\text{rel. loss of acc. } (\bar{x}_n))^2 + \ldots + (\text{rel. loss of acc. } (\bar{x}_{r+1}))^2]^{1/2}}{n + 1 - r},$$

where $\bar{x}_{n+1} = 0, r = n(-1)1.$

The relative loss of accuracy of $(u.y)$ is defined by

$$w_r = q_r \times s_r \times t_r, r = 1(1)n,$$

whilst that of $\bar{x}_r$ is defined by the quantity h_r, where

$$h_r = p_r \times w_r, r = 1(1)n,$$

provided $p_r \times w_r > 10 \times 2^{-t}$, 2^{-t} representing the least significant digit in the word of a binary digital computer. If, however, the above side condition is not satisfied then $h_r = p_r$, if $p_r < w_r$; otherwise $h_r = w_r$. In such a case it seems more appropriate that $h_r = 0{\cdot}1 \times p_r$ or $h_r = 0{\cdot}1 \times w_r$, since in either case w_r or p_r must have some bearing on the evaluation of h_r.

Finally, the quantity h_r, namely the measure of the relative loss of accuracy of $\bar{x}_r$, should satisfy

$$h_r \times (\text{relative error } (\bar{x}_r)) \doteqdot 2^{-t}, r = 1(1)n. \qquad (2)$$

In the ALGOL procedure that follows no pivoting has been used and consequently the process can break down even when A is very well-conditioned: for example

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Even more serious (because it happens far more frequently) it may be numerically unstable when A is well-conditioned: for example

$$A = \begin{bmatrix} \epsilon & 1 \\ 1 & \epsilon \end{bmatrix},$$

where $\epsilon = 0(10^{-10})$. In fact the process is unnecessarily unstable whenever a principal submatrix of A is much more ill-conditioned than A itself. These 'unnecessary' failures and instability may, usually, be avoided by partial pivoting (see, for example, Wilkinson, 1963 and 1965).

Nevertheless, the emphasis here is not on the solution of (1), but on the empirical computation of the component-wise relative loss of accuracy of $\bar{x}$. Certainly one can easily introduce partial pivoting into the procedure following closely the algorithm given by Bowdler, Martin, Peters and Wilkinson (1966).

3. ALGOL procedure

Formal parameter list:

n order of the matrix *A*.
a elements of the matrix *A* stored as $n \times n$ array.
b elements of *b* stored as $n \times 1$ array.
h elements of the computed relative loss of accuracy of $\bar{x}$ stored as $n \times 1$ array.

```
procedure solve G (a,b,h,n); value n; array a,b,h; integer n;
comment Solves Ax=b. The unsymmetric matrix, A, is
stored in the n×n array a[i,j], i=1(1)n, j=1(1)n, b is
stored in the array b[i], i=1(1)n. The decomposition
A=LU, where L is a unit lower triangular matrix and U
an upper triangular matrix, is performed and overwritten
on A, omitting the unit diagonal of L. The solution vector
is overwritten on b, whilst the computed relative loss of
accuracy is stored in the array h[i], i=1(1)n. The method
will fail if A, modified by the rounding errors, is singular
or if the pivot is zero or almost zero, since no pivoting
(whether it be partial or total) is used;

begin array e[0: n], k,l,u,w,d[1: n]; real m; integer i,j,r;
for i := 1 step 1 until n do
begin u[i] := abs(a[i,i]); m := 0;
    for j := i+1 step 1 until n do m := m+a[i,j]×a[i,j];
    k[i] := sqrt(m+b[i]×b[i])
end;
for r := 1 step 1 until n−1 do
for i := r+1 step 1 until n do
begin a[i,r] := a[i,r]/a[r,r]; b[i] := b[i]−a[i,r]×b[r];
    m := 0;
    for j :=i+1 step 1 until n do
    begin a[i,j] :=a[i,j]−a[i,r]×a[r,j];
        if j>i then m :=m+a[i,j]×a[i,j]
    end;
    m := l[i] := sqrt(m+b[i]×b[i]);
    if k[i]<m then k[i] := m;
    m := abs(a[i,i]);
    if u[i]<m then u[i] := m
end; l[1] := k[1];
for i := 1 step 1 until n do
begin m := abs(a[i,i]);
    u[i] := if m<u[i] then m/u[i] else 1;
    k[i] := if l[i]<k[i] then l[i]/k[i] else 1
end;
for i := n step −1 until 1 do
begin m := b[i]; if i≠n then
    for j : =i+1 step 1 until n do m := m−a[i,i]×b[j];
    d[i] := abs(m); b[i] := m/a[i,i];
    m := 1;
    for r := i+1 step 1 until n do m := m+b[r]×b[r];
    d[i] := d[i]/(l[i]×sqrt(m))
end;
e[n] := 1; m := 1;
for i := n step −1 until 1 do begin
w[i]:=k[i]×d[i]×e[i]; if w[i]×u[i]<0·72₁₀−11 then begin
if u[i]>w[i] then h[i]:=w[i]×0·1 else h[i]:=u[i]×0·1 end
else h[i] := u[i]×w[i]; m := m+h[i]×h[i];
    e[i−1] := sqrt(m)/(n−i+2) end
end;
```

4. Results and discussion

In what follows cond(A) denotes the condition number of the matrix A, namely

$$\text{cond}(A) = \text{lub}(A)\,\text{lub}(A^{-1}) = \| A \| \, \| A^{-1} \|$$

Table 1

$$A = \begin{bmatrix} 5{\cdot}000 & 2{\cdot}100 & 0{\cdot}050 & 0{\cdot}910 & 0{\cdot}871 & 0{\cdot}030 & 0{\cdot}059 & 0{\cdot}006 & 0{\cdot}871 & 0{\cdot}061 & 0{\cdot}005 \\ 0{\cdot}560 & 6{\cdot}200 & 0{\cdot}030 & -0{\cdot}820 & 0{\cdot}090 & 0{\cdot}073 & 0{\cdot}525 & 0{\cdot}305 & -3{\cdot}009 & 0{\cdot}561 & 0{\cdot}003 \\ -2{\cdot}000 & 0{\cdot}050 & 6{\cdot}800 & -0{\cdot}005 & 0{\cdot}008 & 0{\cdot}012 & 0{\cdot}053 & 0{\cdot}007 & 0{\cdot}623 & -0{\cdot}192 & -1{\cdot}520 \\ 3{\cdot}100 & 0{\cdot}003 & 0{\cdot}059 & 7{\cdot}300 & 0{\cdot}053 & -0{\cdot}092 & -0{\cdot}059 & 0{\cdot}001 & 0{\cdot}007 & -0{\cdot}003 & 2{\cdot}250 \\ 4{\cdot}200 & 0{\cdot}058 & 0{\cdot}067 & 0{\cdot}003 & 8{\cdot}900 & 0{\cdot}562 & 0{\cdot}923 & 0{\cdot}005 & 0{\cdot}010 & 0{\cdot}050 & -0{\cdot}007 \\ 3{\cdot}200 & 1{\cdot}500 & 0{\cdot}331 & -0{\cdot}523 & -0{\cdot}008 & 9{\cdot}700 & 0{\cdot}004 & 0{\cdot}001 & -0{\cdot}009 & 1{\cdot}500 & -0{\cdot}098 \\ -2{\cdot}000 & 0{\cdot}050 & 0{\cdot}060 & 0{\cdot}071 & 0{\cdot}075 & -0{\cdot}502 & 10{\cdot}500 & 0{\cdot}002 & 0{\cdot}053 & -0{\cdot}100 & 0{\cdot}005 \\ 1{\cdot}500 & 1{\cdot}600 & 1{\cdot}700 & -1{\cdot}800 & 0{\cdot}062 & -0{\cdot}065 & -0{\cdot}068 & 11{\cdot}700 & -2{\cdot}200 & 0{\cdot}050 & 0{\cdot}001 \\ 0{\cdot}050 & 0{\cdot}003 & 0{\cdot}007 & 0{\cdot}600 & -0{\cdot}700 & -0{\cdot}500 & 0{\cdot}009 & 0{\cdot}012 & 12{\cdot}300 & -0{\cdot}300 & 5{\cdot}000 \\ 1{\cdot}200 & 1{\cdot}350 & -0{\cdot}059 & 0{\cdot}350 & 6{\cdot}030 & -0{\cdot}570 & -0{\cdot}029 & 0{\cdot}635 & 0{\cdot}920 & 13{\cdot}900 & 1{\cdot}000 \\ -4{\cdot}300 & -0{\cdot}600 & 0{\cdot}052 & 0{\cdot}900 & 0{\cdot}300 & 0{\cdot}800 & 0{\cdot}825 & -0{\cdot}920 & -5{\cdot}300 & -2{\cdot}000 & 14{\cdot}900 \end{bmatrix}$$

$b = (13{\cdot}2450,\ -2{\cdot}8925,\ 2{\cdot}9930,\ 28{\cdot}2870,\ 29{\cdot}7075,\ 38{\cdot}5690,\ 34{\cdot}4875,\ 38{\cdot}1775,\ 79{\cdot}4180,\ 97{\cdot}5050,\ 49{\cdot}5855)^T$.

COMPUTED SOLUTION	EXACT SOLUTION	EXP. REL. ERROR	ACTUAL REL. ERROR
0·5000000000	0·5	$(0{\cdot}34)10^{-8}$	0
1·100000000	1·0	$(0{\cdot}39)10^{-9}$	0
1·500000000	1·5	$(0{\cdot}44)10^{-9}$	0
2·000000000	2·0	$(0{\cdot}11)10^{-8}$	0
2·500000000	2·5	$(0{\cdot}61)10^{-9}$	0
3·000000000	3·0	$(0{\cdot}42)10^{-9}$	0
3·500000000	3·5	$(0{\cdot}29)10^{-9}$	0
4·000000000	4·0	$(0{\cdot}16)10^{-9}$	0
4·500000000	4·5	$(0{\cdot}17)10^{-9}$	0
5·000000000	5·0	$(0{\cdot}81)10^{-10}$	0
5·500000000	5·5	$(0{\cdot}73)10^{-11}$	0

Table 2

$A = (a_{ij}) = (i+j-1)^{-1}$, $i, j = 1(1)5$ (Hilbert matrix of order 5).

$b = (1, 1, 1, 1, 1)^T$.

COMPUTED SOLUTION	EXACT SOLUTION	EXP. REL. ERROR	ACTUAL REL. ERROR
5·000000730	5	$(0{\cdot}13)10^{-7}$	$(0{\cdot}15)10^{-6}$
−120·0000125	−120	$(0{\cdot}18)10^{-7}$	$(0{\cdot}10)10^{-6}$
630·0000512	630	$(0{\cdot}29)10^{-6}$	$(0{\cdot}81)10^{-7}$
−1120·000075	−1120	$(0{\cdot}92)10^{-5}$	$(0{\cdot}67)10^{-7}$
630·0000359	630	$(0{\cdot}25)10^{-5}$	$(0{\cdot}57)10^{-7}$

in the case of symmetric norms. In particular, if the underlying vector norm for the lub norm is the euclidean (2-norm) norm, then $\text{cond}(A) = \sigma_1/\sigma_n$ or $\text{cond}(A) = |\lambda_1|/|\lambda_n|$, if A is Hermitian (symmetric for the real case), where

$$\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_n > 0 \text{ and } |\lambda_1| \geqslant |\lambda_2| \geqslant \ldots \geqslant |\lambda_n| > 0$$

are the singular values and eigenvalues of A respectively. $|A|$ denotes the matrix whose elements are the absolute values of the elements of the matrix A. Also, whenever 0 appears in the column 'Actual Rel. Error', it is inferred that the actual relative error is bounded by 2^{-t}, and for the Atlas computer on which the calculations were performed, $2^{-t} = 2^{-37} \doteq (7{\cdot}275957614)10^{-12}$, taking into consideration the 'noise-level', although the mantissa on Atlas has 40 binary digits including the sign digit.

Table 3

$A = L^{-1}HL$, $H = (h_{ij}) = (i+j-1)^{-1}$, $i, j = 1(1)6$.

$b = (38{\cdot}43333333,\ -6{\cdot}933333334,\ 2{\cdot}307142857,\ -1{\cdot}069841270,\ 0{\cdot}6126984129,\ -0{\cdot}4038961034)^T$.

COMPUTED SOLUTION	EXACT SOLUTION	EXP. REL. ERROR	ACTUAL REL. ERROR
0·9999999575	1	$(0{\cdot}66)10^{-8}$	$(0{\cdot}42)10^{-7}$
2·000001487	2	$(0{\cdot}34)10^{-8}$	$(0{\cdot}74)10^{-6}$
2·999986025	3	$(0{\cdot}35)10^{-6}$	$(0{\cdot}47)10^{-5}$
4·000069004	4	$(0{\cdot}13)10^{-3}$	$(0{\cdot}17)10^{-4}$
4·999764560	5	$(0{\cdot}78)10^{-4}$	$(0{\cdot}47)10^{-4}$
6·000635130	6	$(0{\cdot}10)10^{-2}$	$(0{\cdot}11)10^{-3}$

Table 1 is for a well-conditioned matrix, whilst **Tables 2** and **3** are for very ill-conditioned matrices. In particular, Table 3 is for the Hilbert matrix of order 6 pre- and post-multiplied by

$$L^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ -1 & 3 & -3 & 1 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 \\ -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix}$$

and

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 \\ 1 & 5 & 10 & 10 & 5 & 1 \end{bmatrix}$$

respectively. Table 3 illustrates the well-known result that a small residual vector, in this case (0·0000000000, −0·0000000001, −0·0000000000, 0·0000000000, −0·0000000000, 0·0000000000), does not necessarily imply a highly accurate computed solution vector $\bar{x}$. On the other hand **Table 4**, again for a very ill-conditioned matrix, illustrates the fact that a large residual vector, in this case (0·0000000000, 0·0000000000, 0·0000610352, 0·0312500000), does not necessarily imply an inaccurate computed solution vector $\bar{x}$.

In conclusion, the method is obviously more empirical rather than mathematical in the strict analytic sense. It must be emphasised again that the experimental relative error is not an upper bound of the actual relative error in $\bar{x}$; it does, however, give a good approximation to the order of magnitude of the actual relative error. The tables given are only a small cross-section of many more examples that have been tested and yielded, in general, good results for the computed (experimental) component-wise relative error. The results appear to be more useful than those obtained from the usually quoted formulae (see Bauer, 1963)

$$\frac{|| \bar{x} - A^{-1}b ||}{|| A^{-1}b ||} \leqslant \text{cond}(A) \frac{|| b - A\bar{x} ||}{|| b ||}, \qquad (3)$$

and (see, for example, Khabaza, 1964, p. 91)

$$\frac{|| \bar{x} - A^{-}b ||}{|| A^{-1}b ||} \leqslant \text{cond}(A) \frac{|| \delta A ||}{|| A ||}, \qquad (4)$$

Table 4

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 9{\cdot}9 & 99{\cdot}8 & 999{\cdot}7 \\ 1 & 99{\cdot}8 & 9999{\cdot}9 & 999999{\cdot}8 \\ 1 & 999{\cdot}7 & 999999{\cdot}8 & 999999999{\cdot}9 \end{bmatrix}$$

$b = (10,\ 4319,\ 4030199{\cdot}5,\ 4003001999{\cdot}4)^T.$

COMPUTED SOLUTION	EXACT SOLUTION	EXP. REL. ERROR	ACTUAL REL. ERROR
0·9999999353	1	$(0{\cdot}13)10^{-8}$	$(0{\cdot}65)10^{-7}$
2·000000072	2	$(0{\cdot}24)10^{-7}$	$(0{\cdot}35)10^{-7}$
2·999999993	3	$(0{\cdot}82)10^{-8}$	$(0{\cdot}23)10^{-8}$
4·000000000	4	$(0{\cdot}92)10^{-11}$	0

provided $|| A^{-1} || \, || \delta A || \ll 1$, which give upper bounds for the relative error in terms of the condition number and the relative residuum for (3) and in terms of the condition number and the relative pertubation for (4). More important, the method does not require either the computation of the condition number or the residual vector or upper bounds for $|\delta A|$.

Acknowledgements

The author wishes to thank Mr. I. M. Khabaza, Director of the Computer Centre of Queen Mary College, London, for helpful discussions during the preparation of this work, and the Science Research Council for a maintenance grant. He also thanks the referee for his most helpful criticisms. Part of this work is from a Ph.D. Thesis approved by the University of London.

References

BAUER, F. L. (1966). Genauigkeitsfragen bei der Lösung linearer Gleichungssysteme, *Zeitschrift für Angewandte Mathematik und Mechanik*, Vol. 7, p. 409.

BAUER, F. L. (1963). Optimally Scaled Matrices, *Numerische Mathematik*, Vol. 5, p. 73.

BOWDLER, H. J., MARTIN, R. S., PETERS, G., and WILKINSON, J. H. (1966). Solution of Real and Complex Systems of Linear Equations, *Numerische Mathematik*, Vol. 8, p. 217.

CHARTRES, B. A., and GEUDER, J. C. (1967). Computable Error Bounds for Direct Solution of Linear Equations, *J. Assoc. Comp. Mach.*, Vol. 14, p. 63.

FORSYTHE, G. E. (1960). Crout with Pivoting, *Commun. Ass. Comp. Mach.*, Vol. 3, p. 507.

INTERNATIONAL COMPUTERS AND TABULATORS LTD. (1965). The I.C.T. Atlas 1 Computer Programming Manual for Atlas Basic Language (ABL). I.C.T. publication No. CS 348A.

KHABAZA, I. M. (1964). *A Course of Lectures on Matrix Computations*, London: Institute of Computer Science.

WILKINSON, J. H. (1963). *Rounding Errors in Algebraic Processes*, London: Her Majesty's Stationery Office; New Jersey: Prentice Hall.

WILKINSON, J. H. (1965). *The Algebraic Eigenvalue Problem*, London: Oxford Clarendon Press, pp. 189–264.