AN ERROR ANALYSIS OF A METHOD FOR SOLVING MATRIX EQUATIONS

BY

C. C. PAIGE

STAN-CS-72-297

JUNE 1972

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

BIBLIOGRAPHIC DATA SHEET	**1. Report No.** STAN-CS-72-297	**2.**	**3. Recipient's Accession No.** PB-212 300

4. Title and Subtitle
An Error Analysis of a Method for Solving Matrix Equations

5. Report Date June 1972

6.

7. Author(s)
C. C. Paige

8. Performing Organization Rept. No. STAN-CS-72-297

9. Performing Organization Name and Address
Stanford University
Computer Science Department
Stanford, California 94305

10. Project/Task/Work Unit No.

11. Contract/Grant No.
GJ 29988

12. Sponsoring Organization Name and Address
National Science Foundation
Washington D. C.

13. Type of Report & Period Covered
Technical

14.

15. Supplementary Notes

16. Abstracts

Let $B = [L\ 0]Q$ be a decomposition of the m by n matrix B of rank m such that L is lower triangular and Q is orthonormal. It is possible to solve $Bx = b$ using L but not Q in the following manner: solve $Ly = b$, solve $L^T w = y$, and form $x = B^T w$. It is shown that the numerical stability of this method is comparable to that of the method which uses Q . This is important for some methods used in mathematical programming where B is very large and sparse and Q is discarded to save storage.

17. Key Words and Document Analysis. 17a. Descriptors

error analysis
linear equations
mathematical programming

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement Release Unlimited	**19. Security Class (This Report)** UNCLASSIFIED	**21. No. of Pages** 13
	20. Security Class (This Page) UNCLASSIFIED	**22. Price** $3.00

An Error Analysis of a Method for Solving Matrix Equations

by

C. C. Paige

Abstract

Let $B = [L\ 0]Q$ be a decomposition of the m by n matrix B of rank m such that L is lower triangular and Q is orthonormal. It is possible to solve $Bx = b$ using L but not Q in the following manner: solve $Ly = b$, solve $L^T w = y$, and form $x = B^T w$. It is shown that the numerical stability of this method is comparable to that of the method which uses Q. This is important for some methods used in mathematical programming where B is very large and sparse and Q is discarded to save storage.

*Institute of Computer Science, University of London, England. This work was done while Visiting Research Associate at Stanford University under National Science Foundation grant GJ 29988X.

1. Introduction and Insight

For a given m by n real matrix B with rank m, $n \geq m$, and a real m dimensional vector b, the under-determined set of equations

$$Bx = b \tag{1}$$

can be solved as follows. First use the transformations of either Givens or Householder to obtain the decomposition

$$B = [L\ 0]Q = [L\ 0]\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = LQ_1 \tag{2}$$

where L is lower triangular and Q is orthonormal, so that

$$Q_1 Q_1^T = I \ . \tag{3}$$

This could be done for example for small matrices by applying Householder transformations to B^T via the procedure "decompose" in [1]. A solution of (1) is then seen by substitution to be

$$x = Q_1^T L^{-1} b \ , \tag{4}$$

and since this lies in the range of B^T it is orthogonal to the null space of B and so is the solution which minimizes

$$\| x \|_2 = \sqrt{x^T x} \ . \tag{5}$$

This problem arises in important algorithms used in mathematical programming, for example in [2] and [3]. However, in these cases B is usually very large and sparse and because of storage difficulties it is often uneconomical to store and access Q_1. If this is so, the solution can still be obtained by noting that if w is obtained from

$$BB^Tw = LQ_1Q_1^TL^Tw = LL^Tw = b \tag{6}$$

by solving with L and then L^T, then x is given by

$$x = B^Tw \quad , \tag{7}$$

and this can be seen to give the same mathematical result as in (4).

Unfortunately, when such results are obtained on a computer, rounding errors occur; and the two different approaches are likely to give different answers. Sometimes it has been thought that the second result could be disastrously worse than the first, thus negating to a large extent algorithms similar to the one described by (6) and (7). It is the purpose of this note to show that such algorithms are numerically quite satisfactory.

In order to obtain a clear understanding of the problem and what is happening, a simple computation will be examined before carrying out the full analysis. This computation has no practical use other than to clarify the numerical performance of the actual case. Suppose A is a nonsingular square matrix and

$$AA^Tw = b \tag{8}$$

is solved on a floating point computer with precision ϵ to give $\tilde{w}$. How well does $\tilde{x} = A^T\tilde{w}$ approximate $x = A^Tw$? Note that this computation is similar to that in (6) and (7), except that no special advantage is taken of any decomposition here and no error in forming $A^T\tilde{w}$ will be considered. For simplicity assume that $\| A \|_2 = 1$, so that $\chi \equiv \| A^{-1} \|_2$ is the spectral condition number of A.

The set of equations (8) can be solved in two distinct ways. First AA^T could be computed and the resulting positive definite symmetric matrix equation solved, for example using the Cholesky factorization. From the rounding error analysis [4, p. 115, p. 231] it is known that the computed solution $\tilde{w}$ will satisfy

$$(AA^T + E_1)\tilde{w} = b , \qquad \| E_1 \|_2 = \epsilon_1 \leq f(n)\epsilon , \tag{9}$$

where $f(n)$ is some function of n, the dimension of the problem. But this is just

$$A(I + A^{-1}E_1A^{-T})A^T\tilde{w} = b = AA^Tw \quad , \tag{10}$$

so that on multiplying throughout by A^{-1} and taking norms

$$\| x - \tilde{x} \|_2 = \| A^Tw - A^T\tilde{w} \|_2 = \| A^{-1}E_1A^{-T}A^T\tilde{w} \|_2$$

$$\leq \chi\epsilon_1 \| \tilde{w} \|_2 \leq \chi^2\epsilon_1 \| \tilde{x} \|_2 . \tag{11}$$

But just solving $Ax = b$ directly using a reliable method is known to give a bound on the error $\| x - \tilde{x} \|_2$ proportional to $\chi\epsilon \| \tilde{x} \|_2$, so that if $\| \tilde{w} \|_2$ is very large, the above method for solving this equation can lead to a disastrous loss of accuracy.

For the second approach to solving (8) consider solving $Ay = b$ and then solving $A^T w = y$ for w. Using, for example, triangular decomposition with pivoting, this will give a computed solution $\tilde{w}$ satisfying [4, p. 215, p. 248]

$$(A + E_2)(A^T + E_3)\tilde{w} = b , \qquad \| E_i \|_2 = \epsilon_i \leq f(n)a\epsilon , \qquad (12)$$

where a depends on the largest element arising in the decomposition,

$$\therefore \quad Ax = A(I + A^{-1}E_2)(I + E_3 A^{-T})A^T\tilde{w} = b \qquad (13)$$

so that

$$\| x - \tilde{x} \|_2 \leq \chi\epsilon_2 \| \tilde{x} \|_2 + (\epsilon_3 + \chi\epsilon_2\epsilon_3) \| \tilde{w} \|_2$$

$$\leq (\chi\epsilon_2 + \chi\epsilon_3 + \chi^2\epsilon_2\epsilon_3) \| \tilde{x} \|_2 , \qquad (14)$$

and if $\chi\epsilon_i < 1$, the order of magnitude of the error bound is the same as that for the direct solution. This in effect is what happens in solving (6) and computing (7), that is, whenever the square of the condition number occurs in the error bound for the final solution, it is effectively multiplied by the square of the precision. Note that in both the examples just considered a $\chi^2\epsilon$ term will appear in the error bound for $\tilde{w}$, so that the intermediate vector $\tilde{w}$ could have negligible accuracy, but in the second method of solution the final result could still have quite a few accurate figures. The same comments apply to computing (6) and (7); $\tilde{x}$ will not lose as much accuracy as the intermediate result $\tilde{w}$. This is a fairly regular occurrence in numerical computations and needs to be emphasized.

2. Analysis of the Practical Algorithm

For simplicity in the full analysis the multiplicative terms involving the dimensions of the problem will be omitted from the error bounds. These are relatively unimportant and can be found for any particular computation from the literature [4]. Results of rounding error analyses will be quoted from [4] without further reference, and the symbols ϵ_i will indicate non-negative quantities which are just the product of ϵ, the computer precision, and constants dependent only on the dimensions of the problem. It will be assumed that $\| L \|_2 = 1$ in (2) so that $\gamma \equiv \| L^{-1} \|_2$ is the condition number of L for solution of equations.

The computed lower triangular matrix $\tilde{L}$ obtained by applying the orthogonal transformations of either Givens or Householder to B can be shown to satisfy

$$B + E_4 = [\tilde{L}\ 0]\tilde{Q} = \tilde{L}\tilde{Q}_1 ,$$
$$\tilde{Q}_1\tilde{Q}_1^T = I , \qquad \| E_4 \|_2 = \epsilon_4 ; \tag{15}$$

and when this is combined with (2) it follows that

$$\tilde{L} = LQ_1\tilde{Q}_1^{\ T} + E_4\tilde{Q}_1^{\ T} . \tag{16}$$

The computed solution $\tilde{w}$ of $\tilde{L}y = b$, $\tilde{L}^T w = y$ can be shown to satisfy

$$(\tilde{L} + E_1)\ (\tilde{L}^T + E_2)\tilde{w} = b , \qquad \| E_i \|_2 = \epsilon_i , \tag{17}$$

while the formation of the final solution gives

$$\tilde{x} = (B^T + E_3)\tilde{w} , \qquad \| E_3 \|_2 = \epsilon_3 \quad . \tag{18}$$

Equations (15), (17), and (18) describe the rounding errors that occur in the computation. These will now be manipulated to show their effect on the final solution. From (4), (16) and (17) it can be seen that

$$x = Q_1{}^T L^{-1} b = Q_1{}^T L^{-1}(LQ_1\tilde{Q}_1{}^T + E_4\tilde{Q}_1{}^T + E_1)\ (\tilde{L}^T + E_2)\tilde{w} ,$$

$$= Q_1{}^T[Q_1 + L^{-1}(E_4 + E_1\tilde{Q}_1)]\ [\tilde{Q}_1{}^T\tilde{L}^T + \tilde{Q}_1{}^T E_2]\tilde{w} , \tag{19}$$

where use has been made of $\tilde{Q}_1\tilde{Q}_1{}^T = I$. But using (15) and then (18),

$$\tilde{Q}_1{}^T\tilde{L}^T\tilde{w} = B^T\tilde{w} + E_4{}^T\tilde{w}$$

$$= \tilde{x} + E_4{}^T\tilde{w} - E_3\tilde{w} \tag{20}$$

so that (19) becomes

$$x = Q_1{}^T[Q_1 + L^{-1}(E_4 + E_1\tilde{Q}_1)]\ [\tilde{x} + (E_4{}^T - E_3 + \tilde{Q}_1{}^T E_2)\tilde{w}] . \tag{21}$$

Next from (18), using $Q_1Q_1{}^T = I$,

$$Q_1{}^T Q_1\tilde{x} = Q_1{}^T L^T\tilde{w} + Q_1{}^T Q_1 E_3\tilde{w}$$

$$= B^T\tilde{w} + Q_1{}^T Q_1 E_3\tilde{w}$$

$$= \tilde{x} + (Q_1{}^T Q_1 - I)E_3\tilde{w} \quad ,$$

so that (21) gives

$$x - \tilde{x} = (Q_1^T Q_1 - I)E_3\tilde{w} + Q_1 Q_1^T(E_4^T - E_3 + \tilde{Q}_1^T E_2)\tilde{w}$$

$$+ Q_1^T L^{-1}(E_4 + E_1\tilde{Q}_1)\,[\tilde{x} + (E_4^T - E_3 + \tilde{Q}_1^T E_2)\tilde{w}] \;. \qquad (22)$$

The $Q_1^T Q_1 E_3$ terms cancel in this last equation, and since from (18)

$$\tilde{w} = L^{-T}Q_1\tilde{x} - L^{-T}Q_1 E_3\tilde{w} \quad ,$$

which, if $\chi\epsilon_3 < 1$, gives

$$\| \tilde{w} \|_2 \le \frac{\chi\| \tilde{x} \|_2}{1 - \chi\epsilon_3} \qquad , \qquad (23)$$

it can be seen by taking the norm of (22) that

$$\| x - \tilde{x} \|_2 \le [\epsilon_3 + \epsilon_4 + \epsilon_2 + \chi(\epsilon_4 + \epsilon_1)\,(\epsilon_4 + \epsilon_3 + \epsilon_2)]\,\| \tilde{w} \|_2$$

$$+ \chi(\epsilon_4 + \epsilon_1)\,\| \tilde{x} \|_2$$

$$\le \left\{\chi[\epsilon_1 + \epsilon_4] + \frac{\chi[\epsilon_2 + \epsilon_3 + \epsilon_4]\,[1 + \chi(\epsilon_1 + \epsilon_4)]}{1 - \chi\epsilon_3}\right\}\| \tilde{x} \|_2 \;. \qquad (24)$$

Thus if $\chi\epsilon \ll 1$, the bound on the error in $\tilde{x}$ is proportional to $\chi\epsilon$ rather than $\chi^2\epsilon$ as has often been thought. There is then no catastrophic loss of accuracy in computing (6) and (7) rather than (4), and so the algorithms described in [2] and [3] can safely be used.

This analysis applies to the fully determined case as well as to the under-determined case. Of course the analysis can be simplified if the fully determined case is treated alone, but the result will be just the same. Computational tests carried out by Michael Saunders for the fully determined case using leading parts of the Hilbert matrix indicated that (24) was a fairly tight bound. The computations on the same matrices using (4) gave results well within the bounds for this approach, and so these results were in fact better than those obtained by using (6) and (7). Such comparisons have probably helped to form the myth that (6) and (7) produce a $\chi^2\epsilon$ error effect in the solution x.

Acknowledgements

My thanks to Michael Saunders and Walter Murray for bringing this problem to my attention and for providing information on their computations. I also thank Mike Osborne for making me aware of the wide interest in this result, and Phil Gill for useful comments on the write-up.

References

[1] P. Businger and G. H. Golub, "Linear least squares solutions by Householder transformations," Numer. Math. 7 (1965) pp. 269-276.

[2] P. E. Gill and W. Murray, "A numerically stable form of the Simplex Algorithm," Maths. Report No. 87, National Physical Laboratory, Teddington, England, August 1970.

[3] M. A. Saunders, "Large-scale linear programming using the Cholesky factorization," Computer Science Department Report No. CS 252, Stanford University, Stanford, California, January 1972.

[4] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, 1965.

Keywords

error analysis

linear equations

mathematical programming