# An Algorithm to Estimate the Error In Gaussian Elimination Without Pivoting

*Eleanor Chu*
*Alan George*

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

CS-84-21
August 1984

# An Algorithm to Estimate the Error in Gaussian Elimination Without Pivoting*

*Eleanor Chu*
*Alan George*

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Research Report CS-84-21
August 1984

*ABSTRACT*

This paper deals with the problem of estimating the error in the computed solution to a system of equations when that solution is obtained by using Gaussian elimination without pivoting. The corresponding problem, where either partial or complete pivoting is used, has received considerable attention, and efficient and reliable methods have been developed. However, in the context of solving large sparse systems, it is often very attractive to apply Gaussian elimination without pivoting, even though it cannot be guaranteed a-priori that the computation is numerically stable. When this is done, it is important to be able to determine when serious numerical errors have occurred, and to be able to estimate the error in the computed solution. In this paper a method for achieving this goal is described. Some analysis, along with results of a large number of numerical experiments, suggest that the method is both inexpensive and reliable.

* Research supported in part by Natural Sciences and Engineering Research Council Grant A8111.

## 1. Introduction

This paper deals with the problem of estimating the error in the computed solution to the nonsingular system of equations

$$A x = b \quad , \tag{1.1}$$

where the solution is obtained using Gaussian elimination. In the usual circumstances, the procedure involves two steps. Ignoring rounding errors, the first step is the application of Gaussian elimination to $A$ with some form of row and/or column interchange, yielding a triangular factorization

$$PAQ = LU \quad , \tag{1.2}$$

where $P$ and $Q$ are permutation matrices, $L$ is unit lower triangular, and $U$ is upper triangular. The second step involves the solution of the two triangular systems

$$Ly = Pb \tag{1.3}$$

and

$$U\hat{x} = y \quad , \tag{1.4}$$

where $\hat{x} = Q^T x$ .

In practice, rounding errors will occur in both the decomposition step and the solution of the triangular systems, so the *computed solution* $\tilde{x}$ will satisfy a *perturbed system*.

$$(A+E)\tilde{x} = b \quad . \tag{1.5}$$

The error in the computed solution depends on two quantities, the *relative error in* $A$ , denoted by

$$\gamma = \frac{\| E \|}{\| A \|} \quad , \tag{1.6}$$

and the condition number $K(A)$ of $A$ with respect to inversion, defined by

$$K(A) = \| A \| \, \| A^{-1} \| \quad . \tag{1.7}$$

In particular, the error in $x$, compared to the computed solution $\tilde{x}$, satisfies

$$\frac{\| x - \tilde{x} \|}{\| \tilde{x} \|} \leq \gamma K(A) \quad . \tag{1.8}$$

It is well known that (1.8) is quite realistic for almost all perturbations $E$. That is, the right side of (1.8) is rarely a severe overestimate for the relative error in $\| \tilde{x} \|$.

When partial or complete pivoting is used, $\gamma$ is almost always a *modest* multiple of the machine epsilon $\epsilon$, and the main problem in estimating the error in the computed solution is to obtain an inexpensive estimate for $K(A)$. This problem has received extensive study, and reliable methods have been developed [1, 3, 5, 7]. (We assume that the reader is familiar with these references.) Their success implicitly relies on the assumption that the *computed* triangular factors $\hat{L}$ and $\hat{U}$ are accurate, since what is estimated by these methods is $K(\hat{L}\hat{U})$.

The problem we consider in this paper is different in that we cannot be sure that $\gamma$ is small. In the context of solving very large sparse systems of equations, it is attractive to apply Gaussian elimination *without* pivoting ( $P=Q=I$ in (1.2) above). This allows one to

set up a fixed data structure for $\hat{L}$ and $\hat{U}$ in advance of the numerical computation, so that the actual numerical factorization can be very efficient.

Although this approach very often works well in practice, there is always the danger that unacceptably large roundoff errors will occur during the factorization. Thus, it is important, when using this "no-pivoting" strategy, to be able to determine estimates for *both* $K(A)$ and $\gamma$. Our main contribution is to provide an efficient mechanism for estimating $\|F\|$, where

$$A+F = \hat{L}\hat{U} \quad . \tag{1.9}$$

Experience suggests that the major influence on the error in the computed solution is due to $F$, so a reliable estimator for $F$ is all that is required for purposes of estimating $\gamma$. Rounding errors incurred in solving the triangular systems (1.3) and (1.4) are usually of little significance, or at least do not dominate those due to $F$.

Note that the ratio of $K(A+F)$ to $K(A)$ satisfies the following inequality [1]:

$$\frac{(1-\hat{\gamma})(1-2\hat{\gamma}k)}{(1-\hat{\gamma}k)} \leq \frac{K(A+F)}{K(A)} \leq \frac{1+\hat{\gamma}}{1-\hat{\gamma}k}, \tag{1.10}$$

where $\hat{\gamma} = \dfrac{\|F\|}{\|A\|}$, and $k$ denotes $K(A)$. Therefore, $K(A+F)$ will be of the same order of magnitude as $K(A)$ provided $\hat{\gamma}k \leq 0.01$ .

## 2. A-Posteriori Estimation of the Errors in the Factors

In this section we derive a bound on $\| F \|_1$, where $F$ is given in (1.0). Our bound is computed in terms of $A$ and the elements of the *computed* factors $\hat{L}$ $\hat{U}$, and involves only a modest multiple of $Nonz(\hat{L}+\hat{U})$ arithmetic operations where $Nonz(M)$ denotes the number of nonzero elements in the matrix $M$. We use the standard floating point error analysis model

$$fl(x \bullet y) = (x \bullet y)(1+\delta)$$

$$= (x \bullet y)/(1+\delta') \quad ,$$

where $|\delta| \leq \epsilon$, $|\delta'| \leq \epsilon$, and $\epsilon$ is the unit roundoff error in the floating point system being used. Here $\bullet$ denotes any of the operations of $+$, $-$, $\times$, and $\div$, and $fl(x \bullet y)$ denotes the floating point result obtained by applying the machine operation $\bullet$ to the floating point numbers $x$ and $y$.

**Lemma 1.** (Forsythe and Moler [8])

Let $|\delta_i| \leq \epsilon$, $1 \leq i \leq r$, and $r\epsilon \leq 0.01$. Then

$$\prod_{i=1}^{r}(1+\delta_i) = 1+r\theta\epsilon \quad ,$$

where $|\theta| \leq 1.01$.

The elements of $L$ and $U$ are defined respectively by

$$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj})/u_{jj} \quad \text{for } i > j, \tag{2.1}$$

and

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \quad \text{for } i \leq j\ . \tag{2.2}$$

There are several ways of implementing the factorization. For example, there is the conventional row-oriented scheme which involves the sequential creation of zeros below the diagonals. Alternatively, one can implement equations (2.1) and (2.2) directly, which is usually referred to as the Crout decomposition. Sparse matrix techniques generate various other computational schemes. However, the implementation chosen will normally only affect the *order* in which $a_{ij}$ and the elements of the summation enter the computation. For instance, (2.1) may be implemented in any of the ways implied by the parentheses in the equations below.

$$l_{ij} = (a_{ij} - (l_{i1}u_{1j} + (\cdots + (l_{i,j-2}u_{j-2,j} + l_{i,j-1}u_{j-1,j})\ldots)))/u_{jj} \tag{2.3}$$

$$l_{ij} = (a_{ij} - (\cdots ((l_{i1}u_{1j} + l_{i2}u_{2j}) + l_{i3}u_{3j}) \cdots + l_{i,j-1}u_{j-1,j}))/u_{jj} \tag{2.4}$$

$$l_{ij} = (\cdots ((a_{ij} - l_{i1}l_{1j}) - l_{i2}l_{2j}) - \cdots l_{i,j-1}l_{j-1,j})/u_{jj} \tag{2.5}$$

Other orderings of the computation are possible as well, involving different groupings of the summands in the numerator. Similarly, analogous groupings are possible for (2.2).

**Lemma 2.**

Let $s$ be a floating point number defined by the formula

$$s = (c - \sum_{k=1}^{l} a_k b_k)/d \quad , \tag{2.6}$$

where the $l+1$ summands in the numerator are combined (added) in any arbitrary order. (Examples are (2.3), (2.4) and (2.5) ). Then

$$sd + \sum_{k=1}^{l} a_k b_k = c + f \quad , \tag{2.7}$$

where

$$|f| \leq 1.01(l+1)\epsilon(|c| + (\sum_{k=1}^{l} |a_k|\,|b_k|) + |s|\,|d|) \tag{2.8}$$

The proof is left as an exercise.

Let $\hat{L}$ and $\hat{U}$ be the *computed* factors of $A$. Our objective is to derive a bound for $\|F\|_1$, where $\hat{L}\hat{U} = A+F$, as in (1.9). Making the obvious correspondences between (2.1) and (2.6), and between (2.2) and (2.6), Lemma 2 yields

$$\sum_{k=1}^{j} \hat{l}_{ik}\hat{u}_{kj} = a_{ij} + f_{ij} \, , \quad i > j \, , \tag{2.9}$$

and

$$\sum_{k=1}^{i} \hat{l}_{ik}\hat{u}_{kj} = a_{ij} + f_{ij}\ ,\quad i \le j\ , \tag{2.10}$$

where in both cases

$$|f_{ij}| \le 1.01j\epsilon(|a_{ij}| + \sum_{k=1}^{j} |\hat{l}_{ik}|\,|\hat{u}_{kj}|)\quad . \tag{2.11}$$

Note that the summation in (2.11) is simply the $(i, j)$-th element of the product $|\hat{L}|\,|\hat{U}|$, where $|M|$ denotes the matrix whose elements are the absolute values of those of $M$. Thus, (2.11) can be written in matrix form as

$$|F| \le 1.01n\epsilon(|A| + |\hat{L}|\,|\hat{U}|)\quad , \tag{2.12}$$

where the inequality should be understood to apply element by element.

An examination of the elements of $|\hat{L}|\,|\hat{U}|$ shows that $\| |\hat{L}|\,|\hat{U}| \|_1$ is quite inexpensive to compute. Let $\hat{L}_{*i}$ denote the $i$-th column of $\hat{L}$. Then the $j$-th column of $|\hat{L}|\,|\hat{U}|$ is given by $\sum_{i=1}^{j} |\hat{L}_{*i}|\,|\hat{u}_{ij}|$, and $\| |\hat{L}|\,|\hat{U}| \|_1 = \max_j \sigma_j = \sigma$,

where

$$\sigma_j = \sum_{i=1}^{j} \|\hat{L}_{*i}\|_1\,|\hat{u}_{ij}|\quad . \tag{2.13}$$

Note that the computation of the $n$ numbers $\|\hat{L}_{*i}\|_1$ requires about $\frac{n^2}{2}$ additions, and the $\sigma_j$'s together require a further approximately $\frac{n^2}{2}$ multiplications and additions.

Thus, denoting $\alpha = \|A\|_1$, we have as the upper bound for $\frac{\|F\|_1}{\|A\|_1}$, the ratio

$$1.01n\epsilon\frac{(\alpha+\sigma)}{\alpha} \quad . \tag{2.14}$$

The method employed above is very similar to that of Chartres and Geuder [11], although the motivation is different. Their analysis is for a *specific implementation* of the equations (2.1) and (2.2), and their results are therefore somewhat tighter than ours. They also assumed the use of double precision accumulation of inner products, and included the effect of rounding errors incurred in the triangular solution. In contrast, our objective was to have the bound for the error apply to as many *different* computational schemes as possible. In exchange, we obtain a somewhat more pessimistic bound, but for purposes of estimating the error in the solution, this does not seem particularly important. Chartres and Geuder also propose to *explicitly* compute bounds for all elements of $F$, which involves about $\frac{n^3}{3}$ additions, although their basic approach does not preclude using the same technique proposed above if only $\| F \|_1$ is required.

Most error analyses of Gaussian elimination provide a bound for $F$ which involves the number $\rho$, which is the largest number in magnitude that appears in any of the reduced submatrices that occurs during Gaussian elimination. Erisman and Reid [12] propose a scheme for bounding $F$ which involves computing a bound for $\rho$ in terms of the elements of $A$, $\hat{L}$ and $\hat{U}$. Their scheme is no more expensive than ours, but they acknowledge that the bound produced can be quite pessimistic if the elements of $\hat{L}$ and $\hat{U}$ vary greatly in size. The results of Test(3) in section 3 indicate that our scheme appears to behave well, irrespective of the variation in the elements of $\hat{L}$ and $\hat{U}$.

### 3. Numerical Experiments

We have implemented the proposed error estimating scheme for the linear equation solvers in SPARSPAK [9]. Our implementation includes a modified version of the LINPACK condition number estimator and the scheme described above for an a-posteriori estimation of the error in the factors. Both algorithms work on the $LU$ factors obtained from Gaussian elimination without pivoting. The modified condition number estimator has exploited the ideas suggested in [3] and [7] for improved accuracy and efficiency.

Our numerical experiments were designed with the following considerations in mind:

(1) The success of the LINPACK condition number estimation algorithm is based in part on the heuristic that the ill-conditioning of $A$ is inherited in the factor $\hat{U}$ when Gaussian elimination *with* pivoting is used [1]. However, when we apply the same algorithm to the factors obtained via Gaussian elimination *without* pivoting , this situation may be much less likely to occur. Therefore, it is necessary to evaluate the performance of the condition estimator in this new context. That is, we would like to know whether $K'$, the estimate of $K(A+F)$, is still comparable in magnitude to $K(A)$, provided $K(A)\frac{\|F\|}{\|A\|} \leq 0.01$. (Recall that $K(A)\frac{\|F\|}{\|A\|} \leq 0.01$ ensures that $K(A+F)$ is of the same order of magnitude as $K(A)$.)

This question was investigated by comparing $K'$ and $K_L$, where $K_L$ denotes the LINPACK estimate of $K(A)$. The estimate $K_L$ is known to be a good approximation to $K(A)$ in practice, except for some contrived counterexamples [1, 4, 6].

(2) In order to gather direct evidence as to how often the ill-conditioning of $A$ is inherited in $\hat{L}$ instead of $\hat{U}$ when $\hat{L}\hat{U}$ factors are obtained via Gaussian elimination without pivoting, we applied the same condition number estimating algorithm to $\hat{L}^t\hat{U}'$, where

$\hat{L}' = \hat{U}^T$, $\hat{U}' = \hat{L}^T$ and $\hat{L}\hat{U} = A + F$, for the same set of test problems. If $\hat{L}$ is much more ill-conditioned than $\hat{U}$, we would expect the estimate of $K(\hat{L}'\hat{U}')$ to be much larger than the estimate of $K(\hat{L}\hat{U})$.

(3) Recall that in section 2 we have derived an upper bound for the relative factorization error, namely $\frac{\|F\|_1}{\|A\|_1} \leq 1.01n\epsilon\frac{(\alpha+\sigma)}{\alpha}$. Since the upper bound is inevitably pessimistic, it is reasonable to replace it by something more realistic. Our numerical experiments were therefore designed to test whether $\frac{\|F\|_1}{\|A\|_1}$ is reasonably approximated by $\frac{\sigma\epsilon}{\alpha}$ in practice.

(4) We would like to test how well $\frac{\|x - \tilde{x}\|_\infty}{\|x\|_\infty}$ is approximated by $K'\frac{\sigma\epsilon}{\alpha}$ in practice, where $K'$ is the estimate of $K(A+F)$. One should note that the inequality that actually holds is

$$\frac{\|x - \tilde{x}\|_1}{\|\tilde{x}\|_1} \leq K(A)\frac{\|E\|_1}{\|A\|_1},$$

where $(A + E)\tilde{x} = b$,

$$E = F + (\delta\hat{L})\hat{U} + \hat{L}(\delta\hat{U}) + (\delta\hat{L})(\delta\hat{U})$$

$$A + F = \hat{L}\hat{U}$$

$$(\hat{L} + \delta\hat{L})y = b,$$

$$(\hat{U} + \delta\hat{U})\tilde{x} = y.$$

Here $\delta\hat{L}$ and $\delta\hat{U}$ reflect the roundoff errors committed in solving the triangular systems. The inequality we actually tested was instead

$$\frac{\| x - \hat{x} \|_\infty}{\| x \|_\infty} \leq K' \frac{\sigma\epsilon}{\alpha}.$$

Our experiments were performed on a VAX 780. The numerical results reported below address the above four considerations.

Test(1): We generated 99 matrices with density varying from 5% to 100% and nonzero elements uniformly distributed on the interval (0, 1). Of each matrix 20% of its nonzero elements were randomly chosen and scaled by a random factor between $10^4$ and $10^{-10}$. All test matrices were of order 50. The ratio of $\frac{K'}{K_L}$ averaged over 99 matrices was 1.08, while the minimum ratio was 0.16 and the maximum ratio was 2.77.

Test(2): We applied the same condition estimator to the factors $\hat{L}'\hat{U}'$, where $\hat{L}' = \hat{U}^T$, $\hat{U}' = \hat{L}^T$ and $\hat{L}\hat{U} = A + F$. Let $K''$ denote the estimate of $K(\hat{L}'\hat{U}')$. Of the same set of matrices generated for test(1), the ratio $\frac{K''}{K_L}$ averaged over 99 matrices was 1.43, while the minimum ratio was 0.06 and the maximum ratio was 19.6.

A comparison of the ratio $\frac{K''}{K_L}$ to the corresponding ratio $\frac{K'}{K_L}$ in test(1) indicates that the ratio of $\frac{\text{Max}\,(K', K'')}{K_L}$ averaged over 99 tests was 1.62, and the minimum ratio was 0.48. Therefore, if one is prepared to pay the price of executing the condition estimator code twice, a better estimate will be Max $(K', K'')$.

Test(3): For the same set of matrices generated for Test(1) and (2), we computed the *actual* factorization error $\frac{\| F \|_1}{\| A \|_1}$ in the following way:

1. Decompose $A$ via Gaussian elimination without pivoting in single precision.

2. Compute the product of $A' = \hat{L}\hat{U}$ in double precision, where $\hat{L}$ and $\hat{U}$ are the factors obtained from step 1.

3. Compute $F = A' - A$ in double precision.

4. Compute $\frac{\| F \|_1}{\| A \|_1}$.

The actual $\frac{\| F \|_1}{\| A \|_1}$ was then compared to its estimate $\frac{\sigma\epsilon}{\alpha}$. The ratio of $\frac{\sigma\epsilon}{\alpha}$ to $\frac{\| F \|_1}{\| A \|_1}$, i.e., $\frac{\sigma\epsilon}{\| F \|_1}$, averaged over 99 problems was 2.03, while the minimum ratio was 0.87 and the maximum ratio was 3.5.

Test(4): We computed the ratio

$$R = \frac{K' \frac{\sigma\epsilon}{\alpha}}{\frac{\| x - \tilde{x} \|_\infty}{\| x \|_\infty}}$$

by running SPARSPAK and

$$R_L = \frac{K_L \cdot \epsilon}{\frac{\| x - \tilde{x}_L \|_\infty}{\| x \|_\infty}}$$

by running LINPACK for the following test problems. For each problem, we generated an appropriate righthand side by assuming that the solution was a vector with components all 1's. Since the linear equation solvers in LINPACK implemented Gaussian elimination with partial pivoting, $\frac{\| x - \tilde{x}_L \|_\infty}{\| x \|_\infty} < \frac{\| x - \tilde{x} \|_\infty}{\| x \|_\infty}$ in

general.

1. The ratios $R$ and $R_L$ were computed for 40 positive definite matrices with density varying from 5% to 100%. The test matrices were generated with nonzero off-diagonal elements uniformly distributed on the interval (-1, 1) and each diagonal element equal to the sum of the magnitudes of the off-diagonal elements in each corresponding row. All matrices were of order 50. Among the 40 test problems, four were indicated as having no correct digits by both SPARSPAK and LINPACK. The computed ratios $R$ and $R_L$ averaged over the remaining 36 problems were 26.9 and 7.2 respectively. The minimum $R$ was 0.85, the maximum $R$ was $5.2\times10^2$, whereas the minimum $R_L$ was 0.46, and the maximum $R_L$ was $1.7\times10^2$.

2. A set of 98 indefinite matrices of order 50 were generated with density varying from 5% to 100% and nonzero elements distributed on the interval (-1, 1). Of each matrix 20% of its nonzero elements were randomly chosen and scaled by a random factor between $10^4$ and $10^{-10}$. Among the 98 test problems, we discarded 10 problems because $K'\frac{\sigma\epsilon}{\alpha} > 0.01$. For six badly scaled problems as reported below, both $K'\frac{\sigma\epsilon}{\alpha}$ and $K_L\cdot\epsilon$ severely overestimate the error in the computed solution.

| *Density* | *Scale* | $R$ | $R_L$ |
|---|---|---|---|
| 0.65 | .1(+5) | .28(+6) | .38(+5) |
| 0.05 | .1(−5) | .15(+9) | .12(+7) |
| 0.05 | .1(−5) | .62(+7) | .20(+5) |
| 0.05 | .1(−5) | .45(+6) | .42(+5) |
| 0.10 | .1(−5) | .85(+7) | .14(+7) |
| 0.10 | .1(−3) | .16(+6) | .34(+5) |

For four other badly scaled problems, SPARSPAK severely overestimated the error, but LINPACK did not:

| *Density* | *Scale* | $R$ | $R_L$ |
|---|---|---|---|
| 0.25 | .1(+5) | .39(+5) | .24(+1) |
| 0.05 | .1(−5) | .98(+7) | .34(+2) |
| 0.05 | .1(−5) | .41(+6) | .12(+2) |
| 0.05 | .1(−5) | .39(+4) | .33(+1) |

The computed ratios $R$ and $R_L$ averaged over the remaining 78 problems were 94 and 3.5 respectively. Among these 78 problems, the minimum $R$ was 6.5, the maximum $R$ was $6.7 \times 10^2$, whereas the minimum $R_L$ was 0.68, and the maximum $R_L$ was 16.

3. The ratio $R$ was also computed for another set of 236 test problems, of which 99 were from test (1) - (3), 110 were indefinite sparse matrices of order 12 to 78 generated with finite element nonzero structures specified in [10], and 27 were large sparse indefinite matrices of order 936 and 1009 generated with finite element nonzero structures specified in [10]. The 137 matrices in

addition to the original 99 used in tests (1) - (3) had 10% to 20% of their nonzero elements ranging in magnitude from $10^4$ to $10^{-10}$. Among the 236 test problems, we discarded 22 problems because $K'\frac{\sigma\epsilon}{\alpha} > 0.01$. For the remaining 214 problems, $K'\frac{\sigma\epsilon}{\alpha}$ appeared to be a reasonable estimate of the maximum relative error in the computed solution when the problem was not badly scaled.

## 4. Conclusions

The modified version of the LINPACK condition number estimator we have implemented for SPARSPAK [9] performs well on randomly generated sparse indefinite matrices even for the factors obtained from Gaussian elimination without pivoting. We observed no severe underestimation in any of our tests. Of course it should be emphasized that the estimate should not be used if $K'\frac{\sigma\epsilon}{\alpha} > 0.01$, because $K(A+F)$ may then be significantly different from $K(A)$, and $K'$, the *estimate* of $K(A+F)$, may have very little connection with $K(A)$.

If one is prepared to execute the condition estimator twice, a better estimate of $K(A)$ can be obtained from Max $(K', K'')$, where $K'$ is the estimate of $K(\hat{L}\hat{U})$, $A + F = \hat{L}\hat{U}$, and $K''$ is the estimate of $K(\hat{L}'\hat{U}')$, where $\hat{L}' = \hat{U}^T$, and $\hat{U}' = \hat{L}^T$. However, since no unreasonable underestimation was observed when comparing $K'$ to $K_L$ throughout our tests, we chose not to compute $K''$ in our SPARSPAK implementation [9].

The results of Test(3) show that the actual relative error in the factors

appears to be very reliably estimated by the ratio $\frac{\sigma\epsilon}{\alpha}$.

The maximum relative error in the computed solution appears to be bounded by the product $K'\frac{\sigma\epsilon}{\alpha}$ and this estimate is rarely a severe overestimate of $\frac{\| x-\tilde{x} \|_\infty}{\| x \|_\infty}$ for well-scaled problems. No underestimation was observed throughout our tests, although it may very well be possible to contrive examples for which $K'\frac{\sigma\epsilon}{\alpha}$ underestimates the relative error.

**References**

[1] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, *An estimate for the condition number of a matrix,* SIAM J. Numer. Anal. *16* (1979), pp. 368-375.

[2] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK Users' Guide,* Society for Industrial and Applied Mathematics, Philadelphia, 1979.

[3] D. P. O'Leary, *Estimating Matrix Condition Numbers,* SIAM J. Sci. Stat. Comput., *1* (1980), 205-209.

[4] A. K. Cline and R. K. Rew, *A Set of Counter-Examples to Three Condition Number Estimators,* SIAM J. Sci. Stat. Comput., Vol. 4, No. 4, December 1983, pp. 602-611.

[5] A. K. Cline, A. R. Conn and C. F. Van Loan, *Generalizing the LINPACK Condition Estimator,* Lecture Notes in Mathematics, vol. 909, Springer-Verlag, 1982, pp. 73-83.

[6] D. Heller, *More Counterexamples to the LINPACK Condition Estimator,* Manuscript, Department of Computer Science, The Pennsylvania State University, 1981.

[7] R. G. Grimes and J. G. Lewis, *Condition Number Estimation For Sparse Matrices,* SIAM J. Sci. Stat. Comput., Vol. 2, no. 4, 1981, pp. 384-388.

[8] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems,* Prentice-Hall, Englewood Cliffs, 1967.

[9] E. C. H. Chu, J. A. George, J. W. H. Liu, and E. G. Y. Ng, *SPARSPAK: Waterloo Sparse Matrix Package, User's Guide for SPARSPAK-A, A collection of modules for solving sparse systems of linear equations,* Department of Computer Science, University of Waterloo, 1984.

[10] A. George and J. W-H Liu, *Algorithms for matrix partitioning and the numerical solution of finite element systems,* SIAM J. Numer. Anal. 15 (1978), pp. 297-327.

[11] B. A. Chartres and J. C. Geuder, *Computable Error Bounds for Direct Solution of Linear Equations,* J. ACM, Vol. 14, No. 1, January 1967, pp. 63-71.

[12] A. M. Erisman and J. K. Reid, *Monitoring the Stability of the Triangular Factorization of a Sparse Matrix,* Numer. Math. 22, 183-186(1974).