# The Calculation of Ordinary Least Squares Estimates

by

Robert E. Hall

Department of Economics
University of California, Berkeley

March 1970

## Abstract

The basic point of this paper is that fast, error-prone methods of computation are useful in least squares calculations, provided that information is available about the magnitude of the errors. The main contribution of the paper is a set of bounds on the errors of a certain class of efficient least squares algorithms. These bounds should be useful to the statistician dealing with any but the most seriously ill-conditioned regression problem.

# The Calculation of Ordinary Least Squares Estimates[1]

by

Robert E. Hall

## 1. The Problem

Abundant experimental evidence now exists to demonstrate that a certain amount of care is required in order to obtain accurate results in least squares regressions. The papers of Longley [ 2 ] and Wampler [ 3 ] are particularly relevant for practical statisticians, since they report on the results of experiments with computer programs that are in widespread use in empirical research. Many of the programs performed very poorly on the various test problems of the two authors, and it is reasonable to expect that these programs are undergoing extensive revision in order to improve their accuracy.

Among efficient computer programs, those that are more accurate are also more time-consuming. The dominant theme of the

---

[1]This is a completely revised version of an earlier paper of the same title, issued as Working Paper 130, IBER-CRMS, University of California, Berkeley, March 1968, and as Working Paper 3, Department of Economics, Massachusetts Institute of Technology, July 1967.

recent literature in this area has been an advocacy of more and more accurate algorithms, which are necessarily more and more demanding of computer time. Some truly extravagant algorithms have been proposed and tested, achieving astonishing accuracy in very badly conditioned problems. It would be natural to draw the conclusion that an accurate, expensive algorithm should be adopted for general use in least squares calculations, in order to guard against serious errors in particularly difficult problems. Clearly, however, this is a wasteful strategy. A more economical approach is to use a cheap, rough algorithm for well-conditioned problems and to save the refined and expensive algorithm for problems that actually require it.

The purpose of this paper is to present an analytical basis for the approach just suggested. The main contribution is a set of bounds on the errors in a certain useful class of least squares algorithms. These bounds can be calculated *a posteriori* to evaluate the accuracy of a set of regression results, and a more accurate member of the class of algorithms can be selected if the first set of results are inadequate. The more accurate methods are essentially iterative refinements of the simplest method, and make use of some of its intermediate results. As a consequence, very little effort is lost by calculating results first for the simplest algorithm as a routine procedure, and going on to the refinement only when necessary. There is no need for the user to guess *a priori* how well conditioned his problem may be.

## 2. Approximate Error Bounds for Direct Calculation

Direct solution of the normal equations,

$$X'Xb = X'y \tag{1}$$

(where b is n x 1, X is T x n, and y is T x 1) is the easiest known algorithm for regression calculations. An efficient scheme is the following:

(i) Form the matrix $M = X'X$ and the vector $m = X'y$

(ii) Calculate an upper-triangular matrix, U, such that $UU' = M$, by Cholesky's square root method [4, pp. 305-307]

(iii) Solve the equation $Uz = m$ by elimination

(iv) Solve the equation $U'b = z$ for b.

This scheme takes full advantage of the symmetry of M. The optimal allocation of computer resources generally involves the use of single-precision storage for all arrays and double-precision accumulation of all inner products. This arrangement is assumed throughout the paper.

The basic source of error in these and other computer calculations arises from the fact that numbers cannot be stored internally in exact form, but must be approximated by members of a finite set of numbers that can be expressed in a certain fixed number of binary

digits. We assume that all numbers are expressed in floating point form, with t bits in the fraction part, and that each arithmetic operation produces an exact answer that is rounded when it is stored. The effect of rounding is to introduce a relative error of up to $2^{-t}$ in the result, as compared to the exact result for the previously rounded inputs. This is an optimistic view, in that some machines operate in a cruder fashion.

Our study rests on the backward error analysis of Wilkinson, [ 5 ]. Instead of examining directly the difference between the calculated and true results, we ask the opposite question: For what problem are the computed results exactly correct? Our answer has important implications for regression calculations -- it shows that the errors made in calculating $b = M^{-1}m$ are at most of the same order of magnitude as those caused by rounding the true M to the number of digits stored in the computer. Even if b could be calculated exactly from the rounded M, it would not be significantly more accurate than that calculated in single precision. The practical implication of this observation is that refinement of the calculation of b from the rounded M would be pointless. The simple algorithm outlined above gives results that are close to the theoretical limitation in accuracy imposed by the rounding of M. Higher accuracy can be obtained in algorithms that solve $Mb = m$ only by storing M in

double precision.

The basis for this conclusion is stated more precisely in

Theorem 1

When floating point arithmetic (with relative error $\delta = 2^{-t}$) is used to solve the equation

$$Mb = m \tag{2}$$

where M is n x n and positive definite, and $|M_{ij}| < (1-\delta)\sqrt{M_{ii}M_{jj}}$, the computed solution, b*, satisfies

$$(M + E)b^* = m \quad , \tag{3}$$

where

$$|E_{ij}| \leq 4\delta\sqrt{M_{ii}M_{jj}} \tag{4}$$

Proof:

By reinterpreting a result of Wilkinson [ 4, pp. 305-307], we obtain the following bounds for the Cholesky decomposition:

$$UU' = M + F \tag{5}$$

(6) $$|F_{ij}| \leq \delta\sqrt{M_{ii}M_{jj}}$$

A second result of Wilkinson [5, pp. 103-104] gives

(7) $$(U + G)z^* = m$$

for the computed solution to

(8) $$Uz = m$$

where G is a diagonal matrix whose elements are bounded by

(9) $$|G_{ii}| \leq \delta\, U_{ii} \quad .$$

Applying this result again, we have

(10) $$(U + H)'b^* = z^*$$

for the computed solution to

(11) $$U'b = z^*$$

and H is a diagonal matrix with

(12) $|H_{ii}| \leq \delta U_{ii}$ .

Thus,

(13) $(U + G)(U + H)'b^* = m$

(14) $(UU' + UH' + G'U + GH')b^* = m$

(15) $(M + F + UH' + G'U + GH')b^* = m$ ,

so the matrix E given in the statement of the theorem is

(16) $E = F + UH' + G'U + GH'$

and

(17) $|E_{ij}| \leq |F_{ij}| + |(UH')_{ij}| + |(G'U)_{ij}| + |(GH')_{ij}|$ .

Now

(18) $(UH')_{ij} = U_{ij}H_{jj}$

so

(19) $|(UH')_{ij}| \leq \delta\sqrt{M_{ii}M_{jj}}$ .

The same holds for $|(G'U)_{ij}|$. Finally

$$(20) \qquad |(GH')_{ii}| \leqslant \delta^2 M_{ii}$$

$$\leqslant \delta M_{ii}$$

Thus

$$(21) \qquad |E_{ij}| \leqslant 4\delta\sqrt{M_{ii}M_{jj}}$$

Q.E.D.

Practical experience suggests that this bound overstates the contribution of errors in solving Mb = m relative to those made by rounding M. The latter appear to dominate in most problems, and the 4 in the bound just derived can safely be replaced by a number less than one.

We have shown that numerical errors in solving the normal equations of least squares are sufficiently small that the computed regression coefficients are the true coefficients for a regression problem that is only very slightly different from the true problem. This does not mean that the computed coefficients are anywhere near the true coefficients. The experimental evidence shows that this is far from the case. Our next step is to provide an assessment of the consequences of errors in M and m (arising from either source) on the calculated coefficients. Here we resort to linearization, which

necessarily introduces a lack of rigor, although since $\delta$ is typically a very small number (in the neighborhood of $10^{-8}$), the missing terms in $\delta^2$, $\delta^3$,... are negligible in any but the most pathological cases.

We now state

Theorem 2

If $g_k$ is the linear approximation to the error in the k'th regression coefficient, defined as

$$g_k = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial b_k}{\partial M_{ij}} E_{ij} + \sum_{i=1}^{n} \frac{\partial b_k}{\partial m_i} e_i \tag{22}$$

(where $E_{ij}$ is the error, actual or imputed back from subsequent calculations, in $M_{ij}$ and $e_i$ is the error in $m_i$)

and

$$|E_{ij}| \leqslant N_1 \delta \sqrt{M_{ii} M_{jj}} \tag{23}$$

$$|e_i| \leqslant N_2 \delta \sqrt{M_{ii} m_o} \tag{24}$$

(where $m_o = \sum_{t=1}^{T} y_t^2$),

then

$$|g_k| \leqslant \delta \sqrt{V_{kk}} \left[ \left( \sum_{i=1}^{n} \sqrt{V_{ii}} \sqrt{M_{ii}} \right) \left( N_2 \sqrt{m_o} + N_1 \sum_{j=1}^{n} |b_j| \sqrt{M_{jj}} \right) \right] \tag{25}$$

where $V = M^{-1}$

Proof:

We observe first that

$$\frac{\partial b_k}{\partial M_{ij}} = -V_{ki} b_j \tag{26}$$

so, since V is positive definite,

$$\left|\frac{\partial b_k}{\partial M_{ij}}\right| \leqslant \sqrt{V_{kk} V_{ii}}\ |b_j| \tag{27}$$

and

$$\frac{\partial b_k}{\partial m_i} = V_{ki} \tag{28}$$

so

$$\left|\frac{\partial b_k}{\partial m_i}\right| \leqslant \sqrt{V_{kk} V_{ii}} \quad . \tag{29}$$

Thus

$$|g_k| \leqslant N_1 \delta \sum_{i=1}^{n} \sum_{j=1}^{n} \sqrt{V_{kk} V_{ii}}\ |b_j|\ \sqrt{M_{ii} M_{jj}} + N_2 \delta \sum_{i=1}^{n} \sqrt{V_{kk} V_{ii}}\ \sqrt{M_{ii} m_o} \tag{30}$$

Q.E.D.

From Theorem 1, we see that for direct calculation, $N_1 = 5$ and $N_2 = 1$. No great effort is required to evaluate this expression,

since the quantity in brackets is the same for all coefficients. The inverse, V, and the square roots of its diagonal elements are usually computed in any case, so the extraction of the square roots of the diagonal elements of M and the actual evaluation of this expression are the only extra steps. In practice, the expression is evaluated with the computed coefficients $b_j^*$. If the result is devoted $h_j$, then we know

$$(31) \qquad b_j^* - h_j \leqslant b_j \leqslant b_j^* + h_j \quad ,$$

except for errors introduced by the linear approximation.

Computational experience with these bounds indicate that they do not greatly overstate the actual magnitude of the errors in the calculated regression coefficients. The reason is that the errors introduced by rounding M and m are generally perverse; the statistical cancellation of errors usually observed in matrix computations does not occur in this case.

## 3. Preliminary Transformation of the Data

In ill-conditioned regression problems the bounds just derived will generally suggest the presence of serious errors in the coefficients calculated by direct solution of the normal equations. Then it becomes necessary to employ a more accurate algorithm or to use double-precision storage for the intermediate results, starting with M and m. On machines with fast double-precision arithmetic and sufficient storage, the latter strategy may be optimal. Wampler's experimental results indicate that double precision gives entirely satisfactory results for his problems, using simple elimination algorithms that are numerically similar to the algorithm of Section 2. One difficulty with this approach is that there is no method of evaluation, a posteriori, for coefficients calculated in this way. The bounds of Section 2 cannot be applied (with a smaller $\delta$) because they are crucially dependent on the use of double-precision accumulation of inner products. The extension to the double-precision case would require triple-precision accumulation, which is entirely uneconomical on modern machines.

The rest of this section is devoted to the study of algorithms based on preliminary transformation of the data. The basic idea is the following: Suppose that R is a nonsingular upper triangular matrix that can be represented exactly in floating point with only

t binary digits. Define

$$\tilde{X} = XR \tag{32}$$

Now suppose we calculate the least squares regression coefficients for y and the transformed variables:

$$\tilde{b} = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'y \tag{33}$$

Then it is easily seen that

$$b = R\tilde{b} \tag{34}$$

The value of these formulas derives from the fact that the only error introduced in forming $\tilde{X}$ from X and b from $\tilde{b}$ is in rounding the result. By an astute choice of the matrix, R, we can deal with a new regression problem that demands far less accuracy in M and m, and then transform the coefficients without losing any of the accuracy gained in calculating $\tilde{b}$ rather than b.

Error bounds for the algorithm based on transformation by an arbitrary matrix, R, are given in:

Theorem 3

If the regression coefficients, $\tilde{b}$, for the transformed regression equation $y = \tilde{X}\tilde{b}$ are calculated by the method of Section 2, and the coefficients for the original problem are calculated as $b = R\tilde{b}$, then bounds on the absolute value of the errors in b are

$$h_j = \sum_{i=j}^{n} |R_{ij}| \, \tilde{h}_j \tag{35}$$

where $\tilde{h}_j$ are the bounds of Theorem 2 for the transformed problem, with $N_1 = 8$ and $N_2 = 2$

Proof:

All that needs to be shown is that the elements of $\tilde{M} = \tilde{X}'\tilde{X}$ and $\tilde{m} = \tilde{X}'y$ are within the bounds required by Theorem 2. First,

$$\tilde{M}_{ij} = \sum_{\tau=1}^{T} (\tilde{X}_{\tau i} + f_{\tau i})(\tilde{X}_{\tau j} + f_{\tau j}) \tag{36}$$

where $f_{\tau k}$ is the error introduced by rounding $\tilde{X}_{\tau k}$:

$$|f_{\tau k}| \leq \delta \, |\tilde{X}_{\tau k}| \tag{37}$$

Thus the error in $\tilde{M}_{ij}$, $E_{ij}$, is given by

$$E_{ij} = \sum_{\tau=1}^{T} f_{\tau i}\tilde{X}_{\tau j} + \sum_{\tau=1}^{T} f_{\tau j}\tilde{X}_{\tau i} + \sum_{\tau=1}^{T} f_{\tau i} f_{\tau j} \tag{38}$$

Now since

$$(39)\qquad \sum_{\tau=1}^{T} |\tilde{X}_{\tau i}\tilde{X}_{\tau j}| \leqslant \sqrt{\Big(\sum_{\tau=1}^{T}\tilde{X}_{\tau i}^2\Big)\Big(\sum_{\tau=1}^{T}\tilde{X}_{\tau j}^2\Big)} \leqslant \sqrt{\tilde{M}_{ii}\tilde{M}_{jj}} \quad ,$$

$$(40)\qquad |E_{ij}| \leqslant 3\delta \sqrt{\tilde{M}_{ii}\tilde{M}_{jj}} \quad .$$

A similar argument shows that

$$(41)\qquad |e_i| \leqslant \delta \sqrt{\tilde{M}_{ii}\tilde{m}_o}$$

These are bounds on the errors in $\tilde{M}_{ii}$ and $\tilde{m}_i$ before final rounding, which adds another relative error bounded by $\delta$. Errors in the solution can be imputed to $\tilde{M}_{ij}$ and are bounded in relative magnitude by $4\,\delta$. Thus the hypotheses of Theorem 2 are fulfilled with $N_1 = 8$ and $N_2 = 2$. Q.E.D.

The choice of the matrix R should strike a balance between the value of the improved accuracy obtainable when R is a full (non-sparse) triangular matrix and the cost of calculating R and carrying out the transformation $\tilde{X} = XR$ on the data. We will discuss two methods for calculating R, one involving a sparse R (and therefore an inexpensive transformation), and one involving a full R. The second method provides exceptionally accurate results.

A well-known method for improving the accuracy of regression calculations is to subtract the mean of each right-hand variable from each observation on that variable. This method is valid only when one of the right-hand variables (say the first) is a constant, in which case it can be represented as follows:

$$(42)\quad R=\begin{bmatrix} 1 & -\mu_2 & \cdots & -\mu_n \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

where $\mu_i$ is the mean of $X_{\tau i}$ and $X_{\tau 1} = 1$ for all $\tau$. This transformation produces an unambiguous improvement in the accuracy of every regression coefficient except possibly the first. From the relations

$$(43)\qquad \tilde{M} = R'MR$$

and

$$(44)\qquad V = R\tilde{V}R'$$

and the fact that $\tilde{V}$ is positive definite, it can be shown that

$$(45)\qquad \tilde{M}_{11} = M_{11}$$

(46) $\tilde{M}_{ii} < M_{ii}$ , (generally by many orders of magnitude),

(47) $\tilde{V}_{11} < V_{11}$

and

(48) $\tilde{V}_{ii} = V_{ii}$

Since $\tilde{b}_i = b_i$ for i=2,..., n, it is immediately apparent from Theorem 2 that substantial gains in accuracy can be obtained by subtracting means. This does not imply that a preliminary data transformation of this type is always to be recommended. The calculation of the means and the subsequent subtraction from the data and accumulation of $\tilde{M}$ require two passes over the data. The second pass would be wasteful if the problem were sufficiently well conditioned to permit satisfactory results to be obtained with the one-pass algorithm of Section 2.

Subtraction of means is limited in its application to regressions with constants. As a result, many computer programs give satisfactory results in ill-conditioned problems with constants, but break down completely in equally ill-conditioned problems without constants. The present framework suggests a simple generalization of the method of subtracting means that is applicable to all regression problems. We take

$$(49) \qquad \mu_i = \frac{M_{1i}}{M_{11}} \quad , \; i=2,\ldots, n \quad .$$

If $X_{\tau 1}$ is constant, this is the conventional method; otherwise it is a new method with the same desirable property of reducing $\tilde{M}_{ii}$ $\tilde{V}_{ii}$ for all i. The improvement obtained with this method depends on how closely $X_{\tau 1}$ is correlated with $X_{\tau 2},\ldots, X_{\tau n}$. If all or most of the correlations are high, the error bounds for the new method may be many orders of magnitude better than those for direct calculation.

The error bound of Theorem 2 attains its irreducible minimum when $\tilde{M}$ is a diagonal matrix, that is, when the transformation has the effect of making the columns of $\tilde{X}$ orthogonal. This suggests that an accurate and costly algorithm might be based on the use of a full triangular matrix R to transform the data, with R chosen so as to make $\tilde{M}$ an identity matrix. That is,

$$(50) \qquad \tilde{M} = R'MR = I$$

or

$$(51) \qquad (R')^{-1}R^{-1} = M \quad ,$$

but since $M = UU'$, we conclude that

$$(52) \qquad R = (U^{-1})' \quad .$$

Accurate calculation of this matrix is almost as difficult as the calculation of regression coefficients. Unlike the coefficients themselves, however, there is no great need for an accurate R. Even a rather inaccurate approximation to the true orthogonalizing R is likely to yield a transformed problem that is much better conditioned than the original problem. Since we are equipped to calculate and evaluate results for an arbitrary R, and not just the one that orthogonalizes $\tilde{X}$, there is no harm in using even a rather bad approximation to the desired R.

The useful feature of this choice of R is that $U^{-1}$ is calculated as an intermediate step in the calculation of V in the simple algorithm of section 2. This means that a candidate for R is immediately available if the results of the first calculation are judged inadequate. The second-stage transformation and calculation can be viewed as an iterative refinement of the first results. In the next section we will refer to this procedure as the <u>two-pass orthonormalization algorithm</u>. Its numerical properties are similar to those of the iterated Gram-Schmidt least squares algorithm[2] with two iterations, but it requires only 2 passes through the raw data, while the Gram-Schmidt requires 2n passes. In addition, it requires substantially fewer arithmetic operations.

---

[2] Davis, [1].

## 4. An Example

In Table 1 we present results for a test problem proposed by Wampler:

$$X_{\tau i} = (\tau-1)^{i-1} \quad , i = 1,\ldots, 6, \tau=1,\ldots, 21 \tag{53}$$

$$b_i = 1 \quad , \text{all } i, \tag{54}$$

and

$$y = Xb \tag{55}$$

Using single-precision arithmetic with t=27 (IBM 7094 and Univac 1108), no conventional least squares program was able to achieve usable accuracy for this problem (see Wampler's Table 1 in [3]). This evidence of ill-conditioning is confirmed by our results for direct calculation, presented in the upper left of Table 1. The actual errors are large, but lie within the calculated bounds. The bounds are quite sharp; some of the errors are more than a tenth of their calculated bounds. The two-pass orthonormalization algorithm of section 3 reduces the actual errors (for t=27) by a factor of about 10000, achieving an accuracy that might suffice in some applications. Unfortunately, the bounds are not as sharp for the second algorithm, and these results would have to be rejected in

Table 1

Results for Wampler's
First Test Problem

t = 27

| True coefficient | Direct Calculation | | | Two-pass Orthonormalization | | |
|---|---|---|---|---|---|---|
| | Calculated coefficient | Actual error | Calculated error bound | Calculated coeffiecient | Actual error | Calculated error bound |
| 1.0000 | -23.2903 | -24.2903 | 394.1074 | .9863 | -.0137 | 7.7320 |
| 1.0000 | 54.5911 | 53.5911 | 433.5782 | 1.0028 | .0028 | 6.2701 |
| 1.0000 | -16.7541 | -17.7541 | 143.0566 | .9995 | -.0005 | 1.6656 |
| 1.0000 | 3.8877 | 2.8877 | 18.6305 | 1.0001 | .0001 | .1866 |
| 1.0000 | .8786 | -.1214 | 1.0365 | 1.0000 | .0000 | .0093 |
| 1.0000 | 1.0032 | .0032 | .0206 | 1.0000 | .0000 | .0002 |

t = 36

| | | | | | | |
|---|---|---|---|---|---|---|
| 1.000000 | .968786 | .031214 | .761494 | 1.000004 | .000004 | .015098 |
| 1.000000 | .994278 | .005722 | .836226 | .999986 | .000014 | .012242 |
| 1.000000 | .982876 | .017124 | .275732 | 1.000003 | .000003 | .003252 |
| 1.000000 | 1.001829 | .001829 | .035902 | 1.000000 | .000000 | .000364 |
| 1.000000 | .999888 | .000112 | .001997 | 1.000000 | .000000 | .000018 |
| 1.000000 | 1.000002 | .000002 | .000040 | 1.000000 | .000000 | .000000 |

Note: These results were obtained by simulating lower precision arithmetic (with rounding) on a machine with t=48 (CDC 6400).

normal use of the bounds.

Table 1 also presents results for a somewhat higher level of precision (t=36, typical for machines with 48 bit words). The bounds for the errors in direct calculation are slightly less tight than for t=27, but they still provide a useful indication of the accuracy of the results. Finally, with t=36, the two-pass orthonormalization algorithm yields sufficiently accurate results in this problem for almost any application of the regression coefficients. Once again, the calculated error bounds are less successful in evaluating the accuracy of the orthonomalization.

Finally, in Table 2 we present results for a reasonably well-conditioned problem, also studied by Wampler. The matrix of right-hand variables is the same as in the first problem, but the coefficients are

$$(56) \qquad b_i = 10^{1-i} \quad , i=1,\ldots, 6$$

With fairly high-precision arithmetic (t=36), quite accurate results are obtained by direct calculation.[3] Furthermore, the bounds are sharp enough to give a useful indication of the accuracy of the coefficients.

---

[3]The fact that Wampler's second problem is so much more tractable than the first, in spite of the fact that they share the same X'X, shows how futile it is to attempt to evaluate the accuracy of results purely on the basis of X'X. Yet many computer programs attempt exactly this by calculating the determinant of X'X. More elaborate measures, such as the condition number of X'X, have the same defect.

Table 2

Results for Wampler's
Second Test Problem

t = 36, Direct calculation

| True coefficient | Calculated coefficient | Actual error | Calculated error bound |
|---|---|---|---|
| 1.000000 | .999999 | .000001 | .000016 |
| 0.1000000 | .1000005 | .0000005 | .0000174 |
| 0.01000000 | .00999944 | .00000056 | .00000575 |
| 0.001000000 | .001000064 | .000000064 | .000000749 |
| 0.0001000000 | .0000999963 | .0000000037 | .0000000416 |
| 0.00001000000 | .00001000001 | .00000000001 | .0000000008 |

## 5. Conclusion

The main contribution of this paper is to provide bounds on the errors of least squares calculations. The usefulness of the bounds is not so much to prove the superiority of more advanced algorithms (the usual application of error bounds), but to make it possible to use fast methods of computation, especially direct calculation, in spite of the fact that these methods are known to break down in ill-conditioned problems. We have also proposed a two-pass procedure for handling the more difficult cases by taking advantage of information accumulated in the unsuccessful attempt at direct calculation. The two-pass algorithm is by no means the most accurate least squares algorithm available, but, again, the ready availability of bounds on its errors makes its use appropriate in a wide variety of applications.

## References

[1] P. Davis, "Orthonormalizing Codes in Numerical Analysis," in J. Todd (ed.), Survey of Numerical Analysis (New York: McGraw-Hill, 1962).

[2] J. W. Longley, "An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User," Journal of the American Statistical Association, Vol. 62, No. 319, pp. 819-841, September 1967.

[3] R. H. Wampler, "An Evaluation of Linear Least Squares Computer Programs," Journal of Research of the National Bureau of Standards -- B. Mathematical Sciences, Vol. 73B, No. 2, pp. 59-90, April - June 1969.

[4] J. H. Wilkinson, "Error Analysis of Direct Methods of Matrix Inversion," Journal of the Association for Computing Machinery, Vol. 8, pp. 281-330, 1961.

[5] _____, Rounding Errors in Algebraic Processes (Englewood Cliffs, New Jersey: Prentice-Hall, 1963).