Algorithms for Computing the Sample Variance: Analysis and Recommendations

TONY F. CHAN, GENE H. GOLUB, and RANDALL J. LEVEQUE*

The problem of computing the variance of a sample of N data points $\{x_i\}$ may be difficult for certain data sets, particularly when N is large and the variance is small. We present a survey of possible algorithms and their round-off error bounds, including some new analysis for computations with shifted data. Experimental results confirm these bounds and illustrate the dangers of some algorithms. Specific recommendations are made as to which algorithm should be used in various contexts.

KEY WORDS: Variance; Standard deviation; Shifted data; Round-off errors; Computer algorithms.

1. INTRODUCTION

The problem of computing the variance of a sample of N data points $\{x_i\}$ is one that seems, at first glance, to be almost trivial but can in fact be quite difficult, particularly when N is large and the variance is small. The fundamental calculation consists of computing the sum of squares of the deviations from the mean,

$$S = \sum_{i=1}^{N} (x_i - \bar{x})^2, \tag{1.1a}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{1.1b}$$

The sample variance is then S/N or $S/(N-1)$ depending on the application. The formulas (1.1) define a straightforward algorithm for computing S. This will be called the *standard two-pass algorithm,* since it requires passing through the data twice: once to compute $\bar{x}$ and then again to compute S. This may be undesirable in many applications, for example when the data sample is too large to be stored in main memory or when the variance is to be calculated dynamically as the data is collected.

To avoid the two-pass nature of (1.1), it is standard practice to manipulate the definition of S into the form

$$S = \sum_{i=1}^{N} x_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)^2. \tag{1.2}$$

This form is frequently suggested in statistical textbooks and will be called the *textbook one-pass algorithm.* Unfortunately, although (1.2) is mathematically equivalent to (1.1), numerically it can be disastrous. The quantities Σx_i^2 and $(1/N)\ (\Sigma x_i)^2$ may be very large in practice and will generally be computed with some rounding error. If the variance is small, these numbers should cancel out almost completely in the subtraction of (1.2). Many (or all) of the correctly computed digits will cancel, leaving a computed S with a possibly unacceptable relative error. The computed S can even be negative, a blessing in disguise since this at least alerts the programmer that disastrous cancellation has occurred.

To avoid these difficulties, several alternative one-pass algorithms have been introduced. These include the *updating algorithms* of Youngs and Cramer (1971), Welford (1962), West (1979), Hanson (1975), and Cotton (1975), and the *pairwise algorithm* of the present authors (Chan, Golub, and LeVeque 1979). In describing these algorithms we use the notation T_{ij} and M_{ij} to denote the sum and the mean of the data points x_i through x_i, respectively,

$$T_{ij} = \sum_{k=\mathrm{i}}^{j} x_k, \qquad M_{ij} = \frac{1}{(j-\mathrm{i}+1)} T_{ij},$$

and S_{ij} to denote the sum of squares

$$S_{ij} = \sum_{k=i}^{j} (x_k - M_{ij})^2.$$

For computing an unweighted sum of squares, as we consider here, the algorithms of Welford, West, and Hanson are virtually identical and are based on the updating formulas

$$M_{1,j} = M_{1,j-1} + \frac{1}{j}(x_j - M_{1,j-1}) \tag{1.3a}$$

$$S_{1,j} = S_{1,j-1} + (j-1)(x_j - M_{1,j-1})\left(\frac{x_j - M_{1,j-1}}{j}\right) \tag{1.3b}$$

*Tony F. Chan is Assistant Professor, Department of Computer Science, Yale University, New Haven, CT 06520. Gene H. Golub is Professor and Chairman, Department of Computer Science, Stanford University, Stanford, CA 94305. Randall J. LeVeque is Research Fellow, Courant Institute of Mathematical Sciences, New York University, New York, NY 10012. This work was supported in part by DOE Contract DE-ACO2-81ER10996, Army Contract DAAG29-78-G-0179, and by National Science Foundation and Hertz Foundation graduate fellowships. The article was produced using *TEX*, a computer typesetting system created by Donald Knuth at Stanford.

with $M_{1,1} = x_1$ and $S_{1,1} = 0$. The desired value of S is ultimately obtained as $S_{1,N}$. The updating formulas of Youngs and Cramer are similar:

$$T_{1,j} = T_{1,j-1} + x_j \tag{1.4a}$$

$$S_{1,j} = S_{1,j-1} + \frac{1}{j(j-1)}(jx_j - T_{1,j})^2 \tag{1.4b}$$

with $T_{1,1} = x_1$ and $S_{1,1} = 0$. These two algorithms have similar numerical behavior and are more stable than the textbook algorithm. Note, in particular, that with both of these algorithms $S = S_{1,N}$ is computed as the sum of nonnegative quantities. Cotton's update is no more stable than the textbook algorithm and should not be used (see Chan and Lewis 1979).

The updating formulas (1.4) can be generalized to allow us to combine two samples of arbitrary size. Suppose we have two samples $\{x_i\}_{i=1}^{m}, \{x_i\}_{i=m+1}^{m+n}$ and we know

$$T_{1,m} = \sum_{i=1}^{m} x_i,$$

$$T_{m+1,m+n} = \sum_{i=m+1}^{m+n} x_i,$$

$$S_{1,m} = \sum_{i=1}^{m} (x_i - \frac{1}{m} T_{1,m})^2,$$

$$S_{m+1,m+n} = \sum_{i=m+1}^{m+n} (x_i - \frac{1}{n} T_{m+1,m+n})^2.$$

Then, if we combine all of the data into a sample of size $m + n$, we have

$$T_{1,m+n} = T_{1,m} + T_{m+1,m+n} \tag{1.5a}$$

$$S_{1,m+n} = S_{1,m} + S_{m+1,m+n} + \frac{m}{n(m+n)}\left(\frac{n}{m} T_{1,m} - T_{m+1,m+n}\right)^2. \tag{1.5b}$$

When $m = n$ this reduces to

$$S_{1,2m} = S_{1,m} + S_{m+1,2m} + \frac{1}{2m}(T_{1,m} - T_{m+1,2m})^2. \tag{1.6}$$

This formula forms the basis of the pairwise algorithm. The pairwise summation algorithm for computing the sum of N numbers is well known and can be described recursively by stating that $T_{1,2m}$ shall be computed as

$$T_{1,2m} = T_{1,m} + T_{m+1,2m}$$

with each of the sums on the right side computed in a similar manner. Formula (1.6) defines the analogous pairwise algorithm for computing the variance. This can be implemented in a one-pass manner using only $O(\log N)$ internal storage locations as discussed in Chan, Golub, and LeVeque (1979) and also by Nash (1981). All logarithms in this article are base 2. It can easily be shown that the use of the pairwise summation algorithm reduces relative errors in $T_{1,N}$ from $O(N)$ to $O(\log N)$ as $N \to \infty$. The pairwise variance algorithm can be expected to have the same advantage, as is confirmed numerically.

Incidentally, pairwise summation can be used in implementing (1.1) (both in computing $\bar{x}$ and in forming S) or (1.2) with similar benefits.

Other devices can also be used to increase the accuracy of the computed S. For data with a large mean value $\bar{x}$, experience has shown that substantial gains in accuracy can be achieved by shifting all of the data by some approximation to $\bar{x}$ before attempting to compute S. Even a crude estimate of $\bar{x}$ can yield dramatic improvements in accuracy, so we need not resort to a two-pass algorithm in order to first estimate $\bar{x}$. This is discussed in detail in Section 3. However, when the shift *is* the computed mean and the textbook algorithm (1.2) is then applied to the shifted data, one obtains the *corrected two-pass algorithm*

$$S = \sum_{i=1}^{N} (x_i - \bar{x})^2 - \frac{1}{N}\left(\sum_{i=1}^{N} (x_i - \bar{x})\right)^2. \tag{1.7}$$

Here the first term is simply the two-pass algorithm (1.1a). The second term would be zero in exact computation but in practice is a good approximation to the error in the first term. Note that in this case use of the textbook algorithm does not lead to catastrophic cancellation, since the correction is generally much smaller than the first term. This algorithm was first pointed out to the authors by Professor Å. Björck (1978) who suggested this correction term based solely on the error analysis of the two-pass algorithm (Chan, Golub, and LeVeque 1979). An alternative (and improved) error analysis is given in Section 3.

Initially algorithms for computing the variance were judged solely on the basis of empirical studies (Hanson 1975, West 1979, and Youngs and Cramer 1971). More recently rigorous error bounds have been obtained for many algorithms (Chan, Golub, and LeVeque 1979; Chan and Lewis 1978, 1979). Our aim here is to present a unified survey of error analyses for the previously mentioned algorithms and techniques. Some of this material is believed to be new, particularly the investigation into the effects of shifting the data. Based on this survey, specific recommendations will be made as to which algorithm should be used in various contexts.

2. CONDITION NUMBERS AND ERROR ANALYSIS

Chan and Lewis (1978) first derived the *condition number*, κ, of a sample $\{x_i\}$ (with respect to computing the variance). This condition number measures the sensitivity of S for the given data set. If relative errors of size γ are introduced into the x_i, then the relative change in S is bounded by $\kappa\gamma$. Chan and Lewis showed this to be true up to $O(\gamma^2)$. In fact it is strictly true as noted by van Nes (1979). Physical data almost always has some uncertainty in it, and this uncertainty will be magnified by the factor κ in S. If nothing else, errors are introduced in representing the data on the computer,

and so a value of S computed on a computer with machine accuracy u may have relative errors as large as κu regardless of what algorithm is used. This value κu can be used as a yardstick by which to judge the accuracy of the various algorithms, especially since error bounds that are functions solely of κ, u, and N can often be derived.

If we define the 2-norm of the data by

$$\| x \|_2^2 = \sum_{i=1}^{N} x_i^2,$$

then the condition number for this problem is given by

$$\kappa = \frac{\| x \|_2}{\sqrt{S}} = \sqrt{1 + \bar{x}^2 N / S}. \tag{2.1}$$

When S is small and $\bar{x}$ is not close to zero we obtain the useful approximation

$$\kappa \approx \bar{x} \sqrt{N/S} \qquad \text{(for } S \text{ small, } \bar{x} \text{ nonzero)}, \tag{2.2}$$

which is the mean divided by the standard deviation. We always have $\kappa \geq 1$, and in many situations κ is very large.

Table 1 shows the asymptotic error bounds for the algorithms discussed. These are bounds on the relative error $|(S - \tilde{S})/S|$ in the computed value $\tilde{S}$. Small constant multipliers have been dropped, for clarity. Higher-order terms have also been dropped, but the terms shown dominate the error bounds whenever the relative error is less than 1. The bounds for the textbook algorithm and West's updating are derived by Chan and Lewis (1978). The two-pass error bound including the $N^2\kappa^2u^2$ term (which can dominate in practice) is derived in Chan, Golub, and LeVeque (1979). Bounds for these algorithms using pairwise summation can be found similarly. The pairwise variance algorithm bound is a conjecture based on the form of the error bound for Youngs and Cramer updating and experimental results. The error analysis for the corrected two-pass algorithm is given in Section 3.

Graphs of these bounds are shown in Figures 1 through 8 along with some experimental results. Each plot has κ on the abscissa and the relative error in S on the ordinate. The lower curve in each figure shows the error bound for $N = 64$, the upper curve for $N = 4096$.

Table 1. Error Bounds for the Relative Error $|(S - \tilde{S})/S|$ in the Computed Value $\tilde{S}$. Only the Dominant Terms are Shown, and Small Constant Factors Have Been Suppressed for Clarity

Algorithm	Error Bound
1. textbook	$N\kappa^2u$
2. textbook with pairwise summation	$\kappa^2u \log N$
3. two-pass	$Nu + N^2\kappa^2u^2$
4. two-pass with pairwise summation	$u \log N + (\kappa u \log N)^2$
5. corrected two-pass	$Nu + N^3\kappa^2u^3$
6. corrected two-pass with pairwise summation	$u \log N + \kappa^2u^3 \log^3 N$
7. updating	$N\kappa u$
8. pairwise	$\kappa u \log N$ (conjectured)

The numerical experiments were performed on an IBM 3081 computer at the Stanford Linear Accelerator Center. The data used were provided by a normal random number generator with mean 1 and a variety of different variances $1 \geq \sigma^2 \geq 10^{-13}$. For this choice of the mean, $\kappa \approx 1/\sigma$ (see (2.2)). In each case the results have been averaged over 20 runs. Single precision was used in all of the tests, with machine accuracy $u \approx 5 \times 10^{-7}$. The "correct" answer for use in computing the error was calculated in double precision. The resulting errors are denoted in the figures by the symbols + (for $N = 64$) and × (for $N = 4096$).

The experimental results confirm the general form of the error bounds given in Table 1. In particular the graphs for the two-pass algorithms show how the higher-order terms (such as $N^2\kappa^2u^2$) begin to dominate the error at fairly modest values of κ.

3. COMPUTATIONS WITH SHIFTED DATA

If we replace the original data $\{x_i\}$ by shifted data $\{\tilde{x}_i\}$ defined by

$$\tilde{x}_i = x_i - d \tag{3.1}$$

for some fixed shift d, then the new data has mean $\bar{x} - d$ and S remains unchanged (assuming the $\tilde{x}_i$ are computed exactly). In practice, data with a nonzero mean is frequently shifted by some a priori estimate of the mean before attempting to compute S. This will generally increase the accuracy of the computed S. We analyze this improvement by investigating the dependence of the condition number on the shift. Bounds on $\tilde{\kappa}$, the condition number of the shifted data, are derived for various choices of the shift d. These can then be inserted in place of κ in the bounds of Table 1 to obtain error bounds for each of the algorithms with shifted data.

From the definition of the condition number we have

$$\tilde{\kappa}^2 = 1 + \frac{N}{S}(\bar{x} - d)^2. \tag{3.2}$$

Comparing this with (2.1) we see that $\tilde{\kappa} < \kappa$ whenever $|d - \bar{x}| < |\bar{x}|$, that is, whenever d lies between 0 and $2\bar{x}$. Taking $d = \bar{x}$ gives perfectly conditioned data, $\tilde{\kappa} = 1$. In practice we cannot compute $\bar{x}$ exactly and usually will not even attempt to compute it (except when using a two-pass algorithm). Instead, we use some rough estimate that is easily computed without a separate pass through all of the data.

Frequently a shift d is obtained by simply "eyeballing" the data. Such a technique might be expected to yield an approximation d that is within a few standard deviations of the mean. This is sufficient to give completely satisfactory bounds on $\tilde{\kappa}$. Recall that the standard deviation is $(S/N)^{\frac{1}{2}}$ and suppose that $|\bar{x} - d| < p(S/N)^{\frac{1}{2}}$ for some small p. Then (3.2) gives

$$\tilde{\kappa}^2 < 1 + p^2. \tag{3.3}$$

For example, if d is within one standard deviation of the

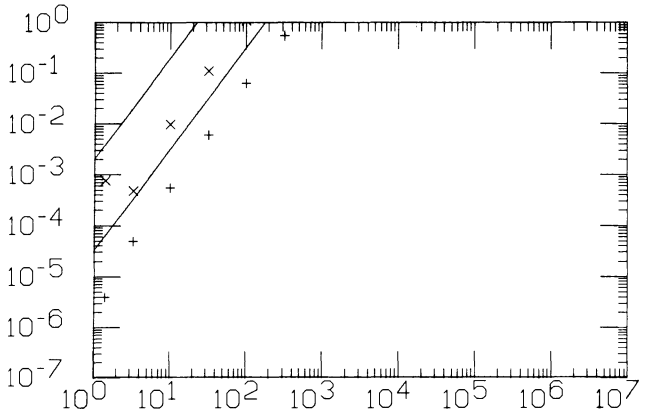

Figure 1. Textbook Algorithm

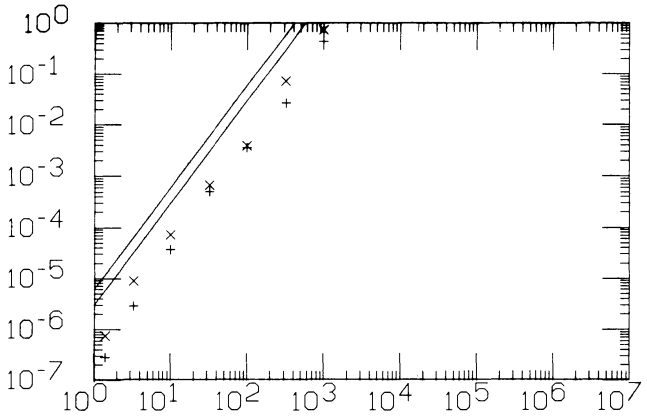

Figure 2. Textbook Algorithm With Pairwise Summation

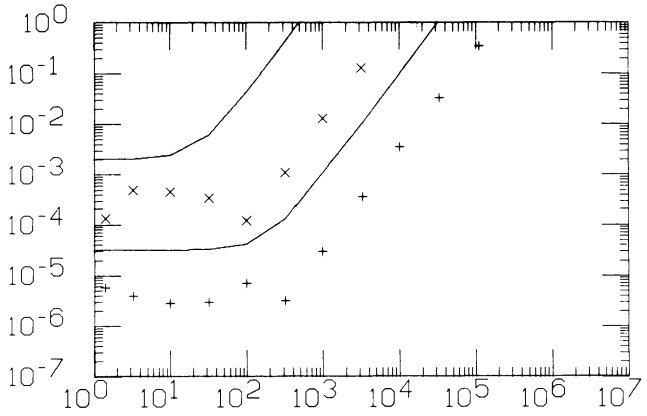

Figure 3. Two-Pass Algorithm

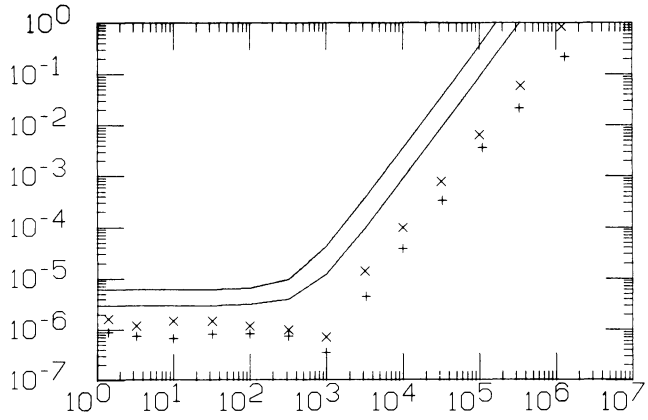

Figure 4. Two-Pass Algorithm With Pairwise Summation

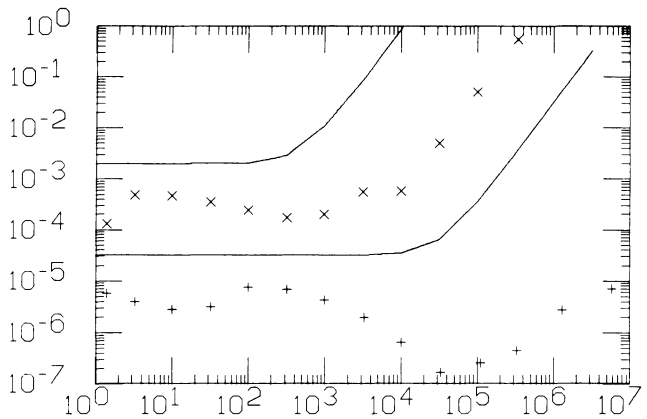

Figure 5. Corrected Two-Pass Algorithm

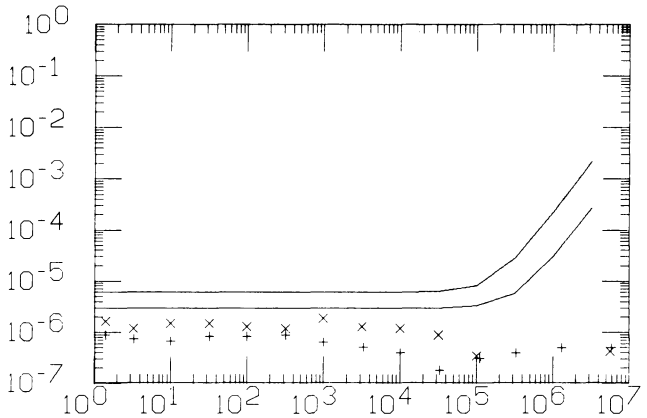

Figure 6. Corrected Two-Pass Algorithm With Pairwise Summation

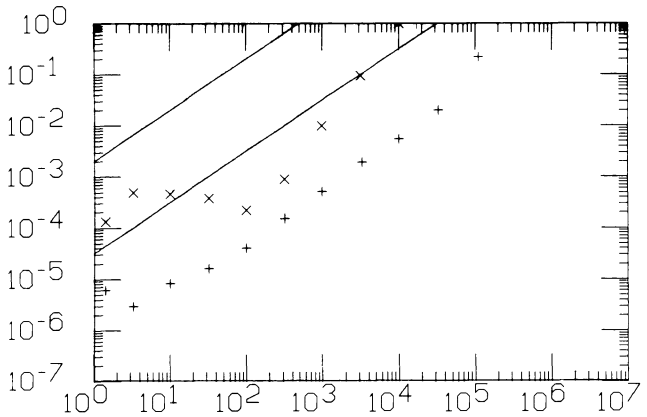

Figure 7. Youngs and Cramer Updating

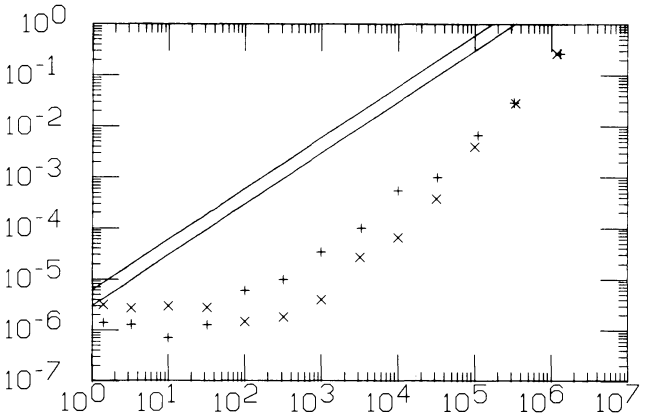

Figure 8. Pairwise Algorithm

mean then $\tilde{\kappa} < \sqrt{2}$. This result is completely independent of S and N.

It is not always possible to obtain an approximation in this manner, nor is it always valid to make such an assumption on its accuracy. Another bound on $\tilde{\kappa}$ can be easily obtained by assuming only that

$$\min_i x_i \le d \le \max_i x_i.$$

This is easly guaranteed, for example by choosing one of the data points as the shift. When $\min x_i \le d \le \max x_i$, we have $(\bar{x} - d)^2 \le \Sigma_i (\bar{x} - x_i)^2 = S$ and so from (3.2),

$$\tilde{\kappa}^2 < 1 + N. \tag{3.4}$$

This bound is not as satisfactory as (3.3), but for moderate values of N it may be sufficient to guarantee acceptable errors in S.

For the case in which we shift by a single data point, $d = x_j$ for some j, we can obtain some interesting probabilistic refinements of (3.4). Equality in (3.4) is unattainable and approximate equality holds only when

$$(\bar{x} - x_j)^2 \approx \sum_i (\bar{x} - x_i)^2,$$

that is, only when x_j lies considerably farther from $\bar{x}$ than do any of the other x_i. If x_j is picked at random from the sample $\{x_i\}$, then the expected value of $\tilde{\kappa}^2$ will be much smaller than $1 + N$. In fact, since $E[(\bar{x} - x_i)^2] = S/N$, (the definition of the sample variance), we have from (3.2) that

$$E[\tilde{\kappa}^2] = 2 \tag{3.5}$$

independent of N and S. Note that this is also independent of the underlying distribution of the $\{x_i\}$. We assumed only that x_j was chosen from $\{x_i\}$ with a uniform distribution. Alternatively we could choose the data value with a fixed index, say x_1, and assume that the data is ordered randomly. This may not be a valid assumption if, for example, initial transients are present in the data.

Improved upper bounds of the form (3.4) that hold with probability close to 1 can also be obtained probabilistically. For fixed k, $1 \le k \le N$, the inequality

$$(\bar{x} - x_i)^2 \ge kS/N$$

can hold for at most N/k values of i. Otherwise we would have $\Sigma(\bar{x} - x_i)^2 > (N/k)(kS/N) = S$. Thus if x_j is chosen at random, there is a probability of at least $(N - N/k)/N = 1 - 1/k$ that $(\bar{x} - x_j)^2 < kS/N$. It follows that

$$\tilde{\kappa}^2 < 1 + k$$

with probability at least $1 - 1/k$ for $1 \le k \le N$. (3.6)

If $N \ge 100$ we have, for example, $\tilde{\kappa}^2 < 101$ with probability .99. This is again independent of N and S when the shift x_j is chosen at random from the sample.

We can generalize this choice of d by using the average of some p data points, $p \ll N$. This average will be denoted by $\bar{x}_p = \Sigma x_j / p$, the sum being over the chosen p data points. We assume that p is sufficiently small that rounding errors in computing $\bar{x}_p$ can be ignored. Specifically this requires $\kappa p u < 1$. The condition number corresponding to this shift is bounded by using Cauchy's inequality,

$$\begin{aligned}\tilde{\kappa}^2 &= 1 + \frac{N}{S}(\bar{x} - \bar{x}_p)^2 \\ &= 1 + \frac{N}{S}\left(\frac{1}{p}\sum_{i=1}^{p}(\bar{x} - x_j)\right)^2 \\ &\le 1 + \frac{N}{Sp}\sum_{i=1}^{p}(\bar{x} - x_j)^2 \\ &\le 1 + \frac{N}{p}.\end{aligned} \tag{3.7}$$

For $p = 1$ this reduces to (3.4). We note that the resulting algorithm can be very easily implemented on a scientific pocket calculator, with great potential for accuracy improvement.

We now consider the case in which the computed mean is used as the shift. In general we cannot ignore rounding errors in computing $\bar{x}$. Instead we compute some approximate floating point value $\mathrm{fl}(\bar{x})$, given by

$$\mathrm{fl}(\bar{x}) = \frac{1}{N}\sum_{i=1}^{N} x_i (1 + \xi_i), \tag{3.8}$$

where the ξ_i are bounded by

$$|\xi_i| \le Nu \tag{3.9}$$

when the usual (forward) summation is used. If pairwise summation is used, the N can be replaced by $\log N$. Now we can bound $\tilde{\kappa}^2$ by

$$\begin{aligned}\tilde{\kappa}^2 &= 1 + \frac{N}{S}(\bar{x} - \mathrm{fl}(\bar{x}))^2 \\ &= 1 + \frac{1}{NS}\left(\sum_{i=1}^{N} x_i \xi_i\right)^2 \le 1 + \frac{1}{NS}\|x\|_2^2 \|\xi\|_2^2 \\ &= 1 + \frac{1}{N}\kappa^2 \|\xi\|_2^2 \le 1 + \kappa^2 \|\xi\|_\infty^2.\end{aligned} \tag{3.10}$$

Here we have used (2.1) and the general inequality $\|\xi\|_2^2 \le N \|\xi\|_\infty^2$, where $\|\xi\|_\infty = \max_i |\xi_i|$. Using (3.9) we can rewrite (3.10) as

$$\tilde{\kappa}^2 \le 1 + N^2 \kappa^2 u^2. \tag{3.11}$$

Note that owing to the dependence on κ, the bound (3.11) may be worse than the bounds obtained for more primitive estimates of d. This reflects situations that can actually occur in practice. One can easily construct examples where the computed mean does not even lie between $\min x_i$ and $\max x_i$ and hence $(\bar{x} - \mathrm{fl}(\bar{x}))^2$ is larger than $\max_i (\bar{x} - x_i)^2$. In this case one is better off shifting by any single data point than by the computed mean.

Of course shifting by the computed mean may also be an undesirable choice from the standpoint of efficiency, since it requires a separate pass through the data to compute $\mathrm{fl}(\bar{x})$. Nonetheless, when a two-pass algorithm is acceptable and $N^2 \kappa^2 u^2$ is small (< 1, say), this shift followed by a one-pass algorithm provides a very dependable method for computing S. The corrected two-pass algorithm (1.7) is of this form; it consists of the textbook algorithm on data shifted by $\mathrm{fl}(\bar{x})$. Its error bound $Nu(1 + N^2 \kappa^2 u^2)$ is easily derived from (3.11) and the textbook algorithm bounds of Table 1.

Other one-pass algorithms could also be used in con-

junction with a shift by the computed mean. However, if a good shift has been chosen so that $\tilde{\kappa} \approx 1$, all one-pass algorithms are essentially equivalent with a bound Nu (or $u \log N$ for algorithms using pairwise summations). Since the textbook algorithm is the most efficient one-pass algorithm (requiring only N multiplications and $2N$ additions as opposed to $4N$ multiplications and $3N$ additions for the updating algorithms, for example), it is the method of choice except in rare instances.

4. RECOMMENDATIONS

The results of the previous sections provide a basis for making an intelligent choice of algorithm for accurately computing the sample variance. First we note that if a parallel processor is available, the data can be split up into smaller samples and the sum of squares computed for each sample individually. These can then be combined, and the global sum of squares computed, by using the updating formulas (1.5). In that case the considerations that follow apply for each processor.

There is one situation in which the textbook algorithm (1.2) can be recommended as it stands. If the data consist only of integers, small enough that no overflows occur, then (1.2) should be used with the sums computed in integer arithmetic. In this case no roundoff errors occur until the final step of combining the two sums, in which a division by N occurs.

For nonintegral data we must first decide whether to use a one-pass or a two-pass algorithm. If all of the data fit in high-speed memory and we are not interested in dynamically updating the variance as new data are collected, then a two-pass algorithm is probably acceptable and the corrected two-pass algorithm (1.7) is recommended. If N is large and high accuracy is needed, it may be worthwhile to use pairwise summation in implementing this algorithm.

If a one-pass algorithm is to be used, the first step is to shift the data as well as possible, perhaps by some x_p as discussed in Section 3. (The probabilistic estimates may be subsequently verified using much tighter a posteriori bounds provided as a by-product of the computation.) Now an appropriate one-pass algorithm must be chosen. We should first estimate $\tilde{\kappa}$, the condition number of the shifted data, perhaps by one of the bounds of Section 3. If $N\tilde{\kappa}^2 u$, the error bound for the textbook algorithm, is at least as small as the desired relative accuracy, then the textbook algorithm can be used on the shifted data. If this bound is too large, we should resort to a less efficient algorithm for safety. The dependence on N can be reduced by the use of pairwise summation. The dependence on $\tilde{\kappa}$ can be reduced by using an updating algorithm. The use of the pairwise algorithm should reduce both of these factors. When N is a power of 2 the pairwise algorithm is fairly easy to implement and requires only $2N$ multiplications and $4N$ additions, which is better than the updating algorithms. For general N slightly more work (particularly human work) is required, making it less attractive.

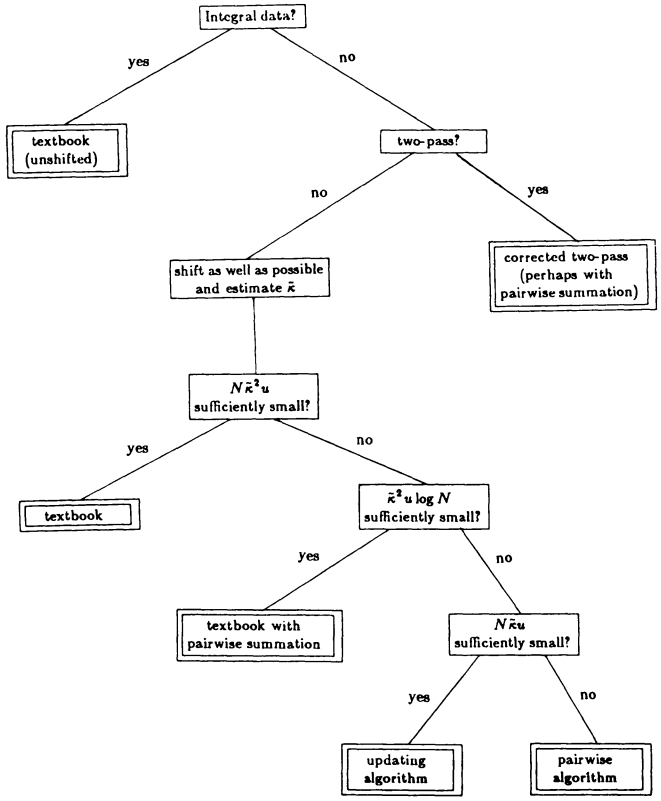

Figure 9. Decision Procedure for Choosing an Algorithm to Compute the Variance. For Details see the Recommendations section

The decision procedure just described is shown graphically in Figure 9.

[*Received April 1982. Revised June 1982.*]

REFERENCES

BJÖCK, Å. (1979), personal communication.

CHAN, T.F., GOLUB, G.H., and LeVEQUE, R.J. (1979), "Updating Formulae and a Pairwise Algorithm for Computing Sample Variances," *Compstat 1982,* Proceedings of the 5th Symposium held at Toulouse, eds. H. Caussinus et al., 30–41.

CHAN, T.F.C., and LEWIS, J.G. (1978), "Rounding Error Analysis of Algorithms for Computing Means and Standard Deviations," Technical Report No. 284, The Johns Hopkins University, Department of Mathematical Sciences.

—— (1979), "Computing Standard Deviations: Accuracy," *Communications of the Association for Computing Machinery,* 22, 526–531.

COTTON, E.W. (1975), Remark on "Stably Updating Mean and Standard Deviation of Data," *Communications of the Association for Computing Machinery,* 18, 458.

HANSON, R.J. (1975), "Stably Updating Mean and Standard Deviation of Data," *Communications of the Association for Computing Machinery,* 18, 57–58.

NASH, J.C. (1981), "Fundamental Statistical Calculations," *Interface Age,* September, 40–42.

VAN NESS, F. (1979), personal communication.

WEST, D.H.D. (1979), "Updating Mean and Variance Estimates: An Improved Method," *Communications of the Association for Computing Machinery,* 22, 532–535.

WELFORD, B.P. (1962), "Note on a Method for Calculating Corrected Sums of Squares and Products," *Technometrics,* 4, 419–420.

YOUNGS, E.A., and CRAMER, E.M. (1971), "Some Results Relevant to Choice of Sum and Sum-of-Product Algorithms," *Technometrics,* 13, 657–665.