Journal of Sciences, Islamic Republic of Iran 21(3): 265-272 (2010) http://jsciences.ut.ac.ir
University of Tehran, ISSN 1016-1104

Camparison of Numerically Stability of Two Algorithms for the Calculation of Variance

D. Rostamy* and E. Yaghesh

Department of Mathematics, Faculty of sciences, University of Imam Khomeini International University, Qazvin, Islamic Republic of Iran

Received: 19 January 2010 / Revised: 7 August 2010 / Accepted: 24 August 2010

Abstract

In descriptive statistics, there are two computational algorithms for determining the variance S^2, of a set of observations $\{x_i\}_{i=1}^n$:

Algorithm 1: $S^2=\frac{1}{n-1}\sum_{i=1}^{n}x_i^2 - \frac{n}{n-1}\bar{x}^2$,

Algorithm 2: $S^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$,

where $\bar{x}=\sum_{i=1}^{n}x_i$. It is interesting to discuss, which of the above formulas is numerically more trustworthy in machine numbers sets. I this paper, based on total effect of rounding error, we prove that the second Algorithm is better than the first Algorithm. Numerical experiments show the efficiency of Algorithm 2.

Keywords: Computational statistics; Round off error; Error analysis

Introduction

The accuracy of results of calculations is a paramount goal in numerical analysis. For computing error analysis in a computational algorithm, sources of error are usually classified as follows:

1. Error in input data,
2. Round off errors, and
3. Approximation errors.

The stability analysis of a numerical method is related to the above sources of error if the amount of the total error is very small [10]. Different strategies for error analysis were investigated by von Neumann and Goldstein [8], Rademacher [9], Scarborough [12], Ashenhurst and Metropolis [1], Wilkinson[16-18], Henrici [4], Moore [7], Kulisch [6], Knuth [5], Sterbenz [14], Bauer and Coworkers [2], [3], and Stauning [13].

Let us denote the set of machine numbers by $\mathbb{F}$ such that (see [11], [14], [15]):

$$\mathbb{F}=\mathbb{F}(\beta,m,L,U)=\{0\}\cup\left\{x\in\mathbb{R}: x=(-1)^s\ \beta^e\sum_{i=1}^{m}\alpha_i\beta^{-i}\right\},$$

where the set of floating point numbers with m significant digits, base $\beta\geq 2$, $\alpha_1\neq 0$, $0\leq\alpha_i\leq\beta-1$, $i=1,\ldots,m$, $s=0,1$ and range of (L,U) with $L\leq e\leq$

* Corresponding author, Tel.: +98(021)44802601, Fax: +98(021)44802601, E-mail: rostamy@khayam.ut.ac.ir

U , $e \in \mathbb{Z}$, $L \in \mathbb{Z}$ and $U \in \mathbb{Z}$. Therefore, it is clear that the members of $\mathbb{F}$ are finite and the cardinality of $\mathbb{F}(\beta, m, L, U)$ is

$$Card\ \mathbb{F} = 2(U - L + 1)(\beta - 1)\beta^{m-1} + 1.$$

The elements of $\mathbb{F}$ are called the machine numbers. If x $\in$ $\mathbb{F}$ then we have to define the following mapping:

$$rd : D \to \mathbb{F}, D \subset \mathbb{R},$$

where rd(x)= x(1+ ε), | ε |≤ eps for all x ∈ D and eps = $\left[\frac{\beta}{2}\right] \times \beta^{-m}$ is called the machine precision or the machine epsilon.

On the other hand, we have observed that the results of arithmetic operations as $\pm, \times, \div, \ldots$ cannot be expected to reproduce the arithmetic operations on $\mathbb{F}$.

We recall that if e < L or e > U then we have the definition of underflow and overflow, respectively.

Also, we consider an algorithm such as $y = \varphi(x)$, that x and y are input and output of φ, respectively. Therefore, its sequence of elementary operations gives rise to a decomposition of φ into a sequence of elementary maps

$$\varphi = \varphi^{(r)} \circ \varphi^{(r-1)} \circ \ldots \circ \varphi^{(0)}$$

$$\varphi^{(i)} = D_i \to D_{i+1},\ D_j \subseteq \mathbb{R}^{n_j},$$

$$D_0 = D,\ D_{r+1} \subseteq \mathbb{R}^{n_{r+1}} = \mathbb{R}^m,$$

where characterize the algorithm. An algorithm for computing the function $\varphi : D \to \mathbb{R}^m, D \subseteq \mathbb{R}^n$, for a given $x = (x_1, \ldots, x_n)^T \in D$ corresponds to a decomposition of the map φ into elementary maps φ^i, and leads from $x^{(0)}$:= x via a chain of intermediate results

$$x = x^{(0)} \to \varphi^{(0)}\left(x^{(0)}\right) =$$

$$x^{(1)} \to \ldots \to \varphi^{(r)}\left(x^{(r)}\right) = x^{(r+1)} \cong y,$$

to a approximation of the result y. We assume that every φ^i is continuously differentiable on D_i. Now let us denote ψ^i the "remainder map" by:

$$\psi^{(i)} = \varphi^{(r)} \circ \varphi^{(r-1)} \circ \ldots \circ \varphi^{(i)} : D_i \to \mathbb{R}^m, i = 0, \ldots, r$$

then $\psi^{(0)} \equiv \varphi$ with floating-point arithmetic, input and round off errors will perturb the intermediate result $x^{(i)}$. Considering these perturbations, we finally arrive at the following formula which describes the effect of the input errors Δx and the round off errors α_i on the result $y \cong x^{(r+1)} = \varphi(x)$:

$$\Delta y = \Delta x^{(r+1)} \mathrel{\hat{=}}$$

$$D\varphi(x)\Delta x + D\psi^{(1)}\left(x^{(1)}\right)\alpha_1 + \ldots \qquad (1)$$

$$+ D\psi^{(r)}\left(x^{(r)}\right)\alpha_r + \alpha_{r+1}.$$

The quantity α_{i+1} can be interpreted as the absolute runoff error newly created when φ^i is evaluated in floating-point arithmetic, and the diagonal elements of E_{i+1} can be similarly interpreted as the corresponding relative round off errors. The notation $\hat{=}$ instead of = , which has used occasionally before, is meant to indicate that the corresponding equations are only a first order approximation, they do not take quantities of higher order $\Delta's$ into account.

$$E_{i+1} = \begin{pmatrix} \varepsilon_1 & 0 & 0 & \cdots & 0 \\ 0 & \varepsilon_2 & 0 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \varepsilon_n \end{pmatrix}, \left|\varepsilon_j\right| \le eps,$$

$$\alpha_{i+1} = E_{i+1} x^{(i+1)}.$$

If one selects a different algorithm for calculating the same result $\varphi(x)$, (in other words a different decomposition of φ into elementary maps), then $D\varphi\Delta x$ remains unchanged; the Jacobian of the matrix $D\psi^i$,which measures the propagation of round off, will be different, however, and so the total effect of rounding error will be,

$$D\psi^{(1)}\ \alpha_1 + \ldots + D\psi^{(r)}\ \alpha_r + \alpha_{r+1}. \qquad (2)$$

Definition 1. *An algorithm is called numerically more trustworthy than another algorithm for calculating* $\varphi(x)$ *if, for a given data set x, the total effect of rounding error (2) smaller for the first algorithm compared to that of the second one*.

Outline of this paper is as follows. First we compute $D\varphi\Delta x$ for two algorithms and show that it will be equal, then we compute total effect of rounding error for two algorithms. Then, comparison between two algorithms and experimental results are given.

Computation of $D\varphi\Delta x$ for Two Algorithms

Proposition 1. *The terms $D\varphi.\Delta x$ are unchanged for two algorithms.*

Proof. In Algorithm 1, we have the following statement:

$$\Delta S^2 = D\varphi.\Delta x = \frac{2(x_1-\bar{x})}{n-1}\Delta x_1 + \ldots + \frac{2(x_n-\bar{x})}{n-1}\Delta x_n . \quad (3)$$

If we write,

$$\varphi = S^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2} - n\,\bar{x}^2\right)$$

$$= \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2} - n\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2\right)$$

$$= \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2} - n\,\frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\right)$$

and

$$D\varphi = \frac{1}{n-1}\left(2x_1 - 2\frac{\sum_{i=1}^{n} x_i}{n}, 2x_2 - 2\frac{\sum_{i=1}^{n} x_i}{n}, \ldots, 2x_n - 2\frac{\sum_{i=1}^{n} x_i}{n}\right)$$

$$= \frac{2}{n-1}\left(x_1-\bar{x}, x_2-\bar{x}, \ldots, x_n-\bar{x}\right),$$

Then (1) is true. Also, for Algorithm 2 we can show (1) is true. We consider

$$\varphi = S^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i-\bar{x})^2\right)$$

$$= \frac{1}{n-1}\left(\left(x_1 - \frac{x_1+\ldots+x_n}{n}\right)^2 + \ldots + \left(x_n - \frac{x_1+\ldots+x_n}{n}\right)^2\right)$$

and

$$D\varphi = \frac{2}{n-1}\begin{pmatrix} \left(x_1 - \frac{x_1+\ldots+x_n}{n}\right)\left(1-\frac{1}{n}\right) \\ +\left(x_2 - \frac{x_1+\ldots+x_n}{n}\right)\left(-\frac{1}{n}\right) + \ldots \\ +\left(x_n - \frac{x_1+\ldots+x_n}{n}\right)\left(-\frac{1}{n}\right) \\ \vdots \\ \left(x_1 - \frac{x_1+\ldots+x_n}{n}\right)\left(-\frac{1}{n}\right) \\ +\left(x_2 - \frac{x_1+\ldots+x_n}{n}\right)\left(-\frac{1}{n}\right) + \ldots \\ +\left(x_n - \frac{x_1+\ldots+x_n}{n}\right)\left(1-\frac{1}{n}\right) \end{pmatrix}^T$$

$$= \frac{2}{n-1}\begin{pmatrix} \left(1-\frac{1}{n}\right)(x_1-\bar{x}) + \left(-\frac{1}{n}\right)\sum_{i=2}^{n}(x_i-\bar{x}) \\ \vdots \\ \left(1-\frac{1}{n}\right)(x_n-\bar{x}) + \left(-\frac{1}{n}\right)\sum_{i=1}^{n-1}(x_i-\bar{x}) \end{pmatrix}^T,$$

then we have:

$$\Delta S^2 = D\varphi.\Delta x =$$

$$\frac{2}{n-1}\left[\left(\left(1-\frac{1}{n}\right)(x_1-\bar{x}) + \left(-\frac{1}{n}\right)\sum_{i=2}^{n}(x_i-\bar{x})\right)\Delta x_1 + \ldots + \left(\left(1-\frac{1}{n}\right)(x_n-\bar{x}) + \left(-\frac{1}{n}\right)\sum_{i=1}^{n-1}(x_i-\bar{x})\right)\Delta x_n\right]$$

$$= \frac{2}{n(n-1)}\left[\left((n-1)(x_1-\bar{x}) - \sum_{i=2}^{n}(x_i-\bar{x})\right)\Delta x_1 + \ldots + \left((n-1)(x_n-\bar{x}) - \sum_{i=1}^{n-1}(x_i-\bar{x})\right)\Delta x_n\right]$$

$$= \frac{2}{n(n-1)}\left[(n-1)(x_1-\bar{x}) - (x_1-\bar{x}))\Delta x_1 + \ldots + \left((n-1)(x_n-\bar{x}) - (x_n-\bar{x})\right)\Delta x_n\right]$$

$$= \frac{2}{n(n-1)}\left[n(x_1-\bar{x})\Delta x_1 + \ldots + n(x_n-\bar{x})\Delta x_n\right],$$

hence, proof is completed. □

Computation of the Total Effect of Rounding Error for Algorithms

The aim is to compute variance with the use of

elementary maps. In fact, in each stage only one computation will be done and we will follow algorithms step by step by elementary maps.

If we consider Algorithm 1, then we can state it by the following decomposition

$$x^{(0)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\end{pmatrix}\xrightarrow{\varphi^{(0)}}x^{(1)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\\ x_1^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(1)}}x^{(2)}$$

$$=\begin{pmatrix}x_1\\ \vdots\\ x_n\\ x_1^{\,2}\\ x_2^{\,2}\end{pmatrix}\cdots x^{(n)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\\ x_1^{\,2}\\ \vdots\\ x_n^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(n)}}$$

$$x^{(n+1)}=\begin{pmatrix}x_{1+}x_2\\ x_3\\ \vdots\\ x_n\\ x_1^{\,2}\\ x_2^{\,2}\end{pmatrix}\cdots\ x^{(2n-1)}=\begin{pmatrix}\sum_{i=1}^{n}x_i\\ x_1^{\,2}\\ \vdots\\ x_n^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(2n-1)}}\ x^{(2n)}$$

$$=\begin{pmatrix}\bar{x}\\ x_1^{\,2}\\ \vdots\\ x_n^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(2n)}}$$

$$x^{(2n+1)}=\begin{pmatrix}\bar{x}^2\\ x_1^{\,2}\\ \vdots\\ x_n^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(2n+1)}}\ x^{(2n+2)}=\begin{pmatrix}n\bar{x}^2\\ x_1^{\,2}\\ \vdots\\ x_n^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(2n+2)}}$$

$$x^{(2n+3)}=\begin{pmatrix}n\bar{x}^2\\ x_1^{\,2}+x_2^{\,2}\\ \vdots\\ x_n^{\,2}\end{pmatrix}$$

$$x^{(3n+1)}=\begin{pmatrix}n\bar{x}^2\\ \sum_{i=1}^{n}x_i^{\,2}\end{pmatrix}\xrightarrow{\varphi^{(3n+1)}}\ x^{(3n+2)}=\sum_{i=1}^{n}x_i^{\,2}-n\bar{x}^2\xrightarrow{\varphi^{(3n+2)}}$$

$$x^{(3n+3)}=\frac{1}{n-1}\sum_{i=1}^{n}x_i^{\,2}-n\bar{x}^2.$$

Also, the following decomposition is given for Algorithm 2.

$$x^{(0)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\end{pmatrix}\xrightarrow{\varphi^{(0)}}x^{(1)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\\ x_1+x_2\end{pmatrix}\cdots$$

$$x^{(n-1)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\\ \sum_{i=1}^{n}x_i\end{pmatrix}\xrightarrow{\varphi^{(n-1)}}\ x^{(n)}=\begin{pmatrix}x_1\\ \vdots\\ x_n\\ \bar{x}\end{pmatrix}$$

$$\xrightarrow{\varphi^{(n)}}x^{(n+1)}=\begin{pmatrix}x_1-\bar{x}\\ x_2\\ \vdots\\ x_n\\ \bar{x}\end{pmatrix}\cdots$$

$$x^{(2n)}=\begin{pmatrix}x_1-\bar{x}\\ \vdots\\ x_n-\bar{x}\end{pmatrix}\xrightarrow{\varphi^{(2n)}}x^{(2n+1)}=\begin{pmatrix}(x_1-\bar{x})^2\\ x_2-\bar{x}\\ \vdots\\ x_n-\bar{x}\end{pmatrix}$$

$$x^{(3n)}=\begin{pmatrix}(x_1-\bar{x})^2\\ \vdots\\ (x_n-\bar{x})^2\end{pmatrix}\xrightarrow{\varphi^{(3n)}}$$

$$x^{(3n+1)}=\begin{pmatrix}(x_1-\bar{x})^2+(x_2-\bar{x})^2\\ (x_3-\bar{x})^2\\ \vdots\\ (x_n-\bar{x})^2\end{pmatrix}\cdots$$

$$x^{(4n-1)}=\sum_{i=1}^{n}(x_i-\bar{x})^2\xrightarrow{\varphi^{(4n-1)}}$$

$$x^{(4n)}=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

It is concluded that if we have n observations $\{x_i\}_{i=1}^{n}$ then using Algorithm1 leads to (3n+3) elementary maps and using Algorithm 2 leads to (4n) elementary maps.

Total Effect of Rounding for Algorithm 1

Proposition 2. *A bound for the total effect of rounding error for Algorithm 1 is as follows*:

$$|A|\le\left[2\left(\frac{x_1^{\,2}+\ldots+x_n^{\,2}}{n-1}\right)+2n\bar{x}^2+|y|\right]eps. \qquad (4)$$

Here, we assume A:= Total effect of rounding error for Algorithm 1, such that

$$A = \frac{x_1^{\,2}}{n-1}\,\varepsilon_1 + \ldots + \frac{x_n^{\,2}}{n-1}\,\varepsilon_n - 2n\bar{x}^2\varepsilon_{n+1} + \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2}\right)\varepsilon_{n+2} + y\;\varepsilon_{n+3}.$$

Proof. If we consider Algorithm 1 then for simplicity we can write:

$$x = x^{(0)} = (x_1,\ldots,x_n),$$

$$\varphi^{(0)}\left(x^{(0)}\right) = x^{(1)} = \left(x_1^{\,2},\ldots,x_n^{\,2}\right),$$

$$\varphi^{(1)}\left(x^{(1)}\right) = x^{(2)} = \left(\sum_{i=1}^{n} x_i^{\,2}, \bar{x}\right),$$

$$\varphi^{(2)}\left(x^{(2)}\right) = x^{(3)} = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2}, n\,\bar{x}^2\right).$$

Therefore, we have:

$$\varphi = \varphi^{(2)} \circ \varphi^{(1)} \circ \varphi^{(0)},$$

$$\Delta y = \Delta x^{(3)} \doteq D\varphi(x)\Delta x + D\psi^{(1)}\left(x^{(1)}\right)\alpha_1 + D\psi^{(2)}\left(x^{(2)}\right)\alpha_2 + \alpha_3$$

and

$$x^{(0)} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \xrightarrow{\varphi^{(0)}} x^{(1)} = \begin{pmatrix} x_1^{\,2} \\ \vdots \\ x_n^{\,2} \\ \bar{x} \end{pmatrix} \xrightarrow{\varphi^{(1)}}$$

$$x^{(2)} = \begin{pmatrix} \sum_{i=1}^{n} x_i^{\,2} \\ \bar{x}^2 \end{pmatrix} \xrightarrow{\varphi^{(2)}} x^{(3n+3)} = \frac{1}{n-1}\sum_{i=1}^{n} x_i^{\,2} - n\bar{x}^2.$$

hence,

$$\varphi^{(0)}(x_1,\ldots,x_n) = \begin{pmatrix} x_1^{\,2} \\ \vdots \\ x_n^{\,2} \\ \bar{x} \end{pmatrix},$$

$$\varphi^{(1)}(u,\ldots,u_{n+1}) = \begin{pmatrix} \sum_{i=1}^{n} u_i \\ u_{n+1} \end{pmatrix},$$

$$\varphi^{(2)}(u,v) = \frac{1}{n-1}\left(u - nv^{\,2}\right),$$

$$\psi^{(1)}(u,\ldots,u_{n+1}) = \varphi^{(2)} \circ \varphi^{(1)}(u,\ldots,u_{n+1}) = \varphi^{(2)}\left(\sum_{i=1}^{n} u_i, u_{n+1}\right) = \frac{1}{n-1}\left(\sum_{i=1}^{n} u_i - n u_{n+1}^{\,2}\right),$$

$$D\psi^{(1)}(u,\ldots,u_{n+1}) = \frac{1}{n-1}(1,\ldots,1,-2nu_{n+1}),$$

$$D\psi^{(1)}\left(x^{(1)}\right) = D\psi^{(1)}\left(x_1^{\,2},\ldots,x_n^{\,2},\bar{x}\right) = \frac{1}{n-1}(1,\ldots,1,-2n\bar{x}),$$

$$\psi^{(2)}(u,v) = \varphi^{(2)}(u,v) = \frac{1}{n-1}\left(u - nv^{\,2}\right),$$

$$D\psi^{(2)}(u,v) = \frac{1}{n-1}(1-2nv),$$

$$D\psi^{(2)}\left(x^{(2)}\right) = D\psi^{(2)}\left(\sum_{i=1}^{n} x_i^{\,2}, \bar{x}\right) = \frac{1}{n-1}(1-2n\bar{x}).$$

Moreover, we have:

$$\overline{\varphi^{(0)}\left(x^{(0)}\right)} - \varphi^{(0)}\left(x^{(0)}\right) = \alpha_1 = \left(x_1^{\,2}\varepsilon_1,\ldots,x_n^{\,2}\varepsilon_n, \bar{x}\varepsilon_{n+1}\right)^T,$$

$$\overline{\varphi^{(1)}\left(x^{(1)}\right)} - \varphi^{(1)}\left(x^{(1)}\right) = \alpha_2 = \left(\left(\sum_{i=1}^{n} x_i^{\,2}\right)\varepsilon_{n+2}, 0\right)^T,$$

$$\overline{\varphi^{(2)}\left(x^{(2)}\right)} - \varphi^{(2)}\left(x^{(2)}\right) = \alpha_3 = y\,\varepsilon_{n+3} = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2} - n\bar{x}^2\right)\varepsilon_{n+3}.$$

So, we can write

$$D\varphi(x)\Delta x + D\psi^{(1)}\left(x^{(1)}\right)\alpha_1 + D\psi^{(2)}\left(x^{(2)}\right)\alpha_2 + \alpha_3 = D\varphi(x)\Delta x + \frac{1}{n-1}\sum_{i=1}^{n}(x_i^{\,2}\varepsilon_i - 2n\bar{x}^2\varepsilon_{n+1}) + \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^{\,2}\right)\varepsilon_{n+2} + y\,\varepsilon_{n+3}.$$

Therefore, the proof is completed. □

Total Effect of Rounding Error for Algorithm 2

Proposition 3. *A bound for the total effect of rounding error for Algorithm 2 is as follows*:

$$|B| \le \left[\frac{(x_1-\bar{x})^2}{n-1} + \ldots + \frac{(x_n-\bar{x})^2}{n-1} + |y|\right] eps. \qquad (5)$$

Here, we assume B:= Total effect of rounding error for Algorithm 2, such that

$$B = \frac{(x_1-\bar{x})^2}{n-1}\varepsilon_2 + \ldots + \frac{(x_n-\bar{x})^2}{n-1}\varepsilon_{n+1} + y\,\varepsilon_{n+3}.$$

Proof. If we consider Algorithm 2 then for simplify we can write:

$$x = x^{(0)} = (x_1,\ldots,x_n),$$
$$\varphi^{(0)}\left(x^{(0)}\right) = x^{(1)} = (x_1,\ldots,x_n,\bar{x}),$$
$$\varphi^{(1)}\left(x^{(1)}\right) = x^{(2)} = \left((x_1-\bar{x})^2,\ldots,(x_n-\bar{x})^2\right),$$
$$\varphi^{(2)}\left(x^{(2)}\right) = x^{(3)} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

Therefore, we have:

$$\varphi = \varphi^{(2)}\circ\varphi^{(1)}\circ\varphi^{(0)},$$

$$\Delta y = \Delta x^{(3)} \doteq D\varphi(x)\Delta x + D\psi^{(1)}\left(x^{(1)}\right)\alpha_1 + D\psi^{(2)}\left(x^{(2)}\right)\alpha_2 + \alpha_3$$

and we can conclude

$$x^{(0)} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \xrightarrow{\varphi^{(0)}} x^{(1)} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ \bar{x} \end{pmatrix} \xrightarrow{\varphi^{(1)}}$$

$$x^{(2)} = \begin{pmatrix} (x_1-\bar{x})^2 \\ \vdots \\ (x_n-\bar{x})^2 \end{pmatrix} \xrightarrow{\varphi^{(2)}} x^{(3)} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

Hence, we have

$$\varphi^{(0)}(x_1,\ldots,x_n) = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ \bar{x} \end{pmatrix},$$

$$\varphi^{(1)}(u,\ldots,u_{n+1}) = \begin{pmatrix} (u_1-u_{n+1})^2 \\ \vdots \\ (u_n-u_{n+1})^2 \end{pmatrix},$$

$$\varphi^{(2)}(v_1,\ldots,v_n) = \frac{1}{n-1}\sum_{i=1}^{n} y_i,$$

$$\psi^{(1)}(u_1,\ldots,u_{n+1}) = \varphi^{(2)}\circ\varphi^{(1)}(u,\ldots,u_{n+1})$$
$$= \varphi^{(2)}\left((u_1-u_{n+1})^2,\ldots,(u_n-u_{n+1})^2\right)$$

$$\frac{1}{n-1}\left(\sum_{i=1}^{n}(u_i-u_{n+1})^2\right)$$
$$= \frac{1}{n-1}\left((u_1-u_{n+1})^2+\ldots+(u_n-u_{n+1})^2\right),$$

$$D\psi^{(1)}(u_1,\ldots,u_{n+1}) = \frac{1}{n-1}\big(2(u_1-u_{n+1}),\ldots,$$
$$2(u_n-u_{n+1}),-2(u_1-u_{n+1}),\ldots,-2(u_n-u_{n+1})\big)$$

$$= \frac{2}{n-1}\left(u_1-u_{n+1},\ldots,u_n-u_{n+1},nu_{n+1}-(u_1+\ldots+u_n)\right),$$

$$D\psi^{(1)}\left(x^{(1)}\right) = D\psi^{(1)}(x_1,\ldots,x_n,\bar{x})$$
$$= \frac{2}{n-1}\left(x_1-\bar{x},\ldots,x_n-\bar{x},n\bar{x}-(x_1+\ldots+x_n)\right)$$
$$= \frac{2}{n-1}(x_1-\bar{x},\ldots,x_n-\bar{x},0),$$

$$\psi^{(2)}(v_1,\ldots,v_n) = \varphi^{(2)}(v_1,\ldots,v_n) = \frac{1}{n-1}\sum_{i=1}^{n} y_i,$$

$$D\psi^{(2)}(v_1,\ldots,v_n) = \left(\frac{1}{n-1},\ldots,\frac{1}{n-1}\right),$$

$$D\psi^{(2)}\left(x^{(2)}\right) = D\psi^{(2)}\left((x_1-\bar{x})^2,\ldots,(x_n-\bar{x})^2\right)$$
$$= \left(\frac{1}{n-1},\ldots,\frac{1}{n-1}\right).$$

Moreover, we have:

$$\overline{\varphi^{(0)}\left(x^{(0)}\right)} - \varphi^{(0)}\left(x^{(0)}\right) = \alpha_1 = (0,\ldots,0,\bar{x}\varepsilon_1)^T,$$

$$\overline{\varphi^{(1)}\left(x^{(1)}\right)} - \varphi^{(1)}\left(x^{(1)}\right) = \alpha_2$$
$$= \left((x_1-\bar{x})^2\varepsilon_2,\ldots,(x_n-\bar{x})^2\varepsilon_{n+1}\right)^T,$$

$$\overline{\varphi^{(2)}\left(x^{(2)}\right)} - \varphi^{(2)}\left(x^{(2)}\right) = \alpha_3 = y\,\varepsilon_{n+2}$$
$$= \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i-\bar{x})^2\right)\varepsilon_{n+2}.$$

So, we can write

$$\Delta y \doteq D\varphi(x)\Delta x + D\psi^{(1)}\left(x^{(1)}\right)\alpha_1 + D\psi^{(2)}\left(x^{(2)}\right)\alpha_2 + \alpha_3,$$

$$\Delta y \doteq D\varphi(x)\Delta x + \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i-\bar{x})^2\varepsilon_{i+1}\right) + y\,\varepsilon_{n+2}.$$

Therefore, the proof is completed. □

Corollary 1. *Comparing (4) and (5) we have*:

$$\left[\frac{(x_1-\bar{x})^2}{n-1}+\ldots+\frac{(x_n-\bar{x})^2}{n-1}+|y|\right]$$

$$\leq \left[2\left(\frac{x_1{}^2+\ldots+x_n{}^2}{n-1}\right)+2n\bar{x}^2+|y|\right]$$

This means that total effect of rounding for Algorithm 2 is less than total effect of rounding for Algorithm 1. Therefore, we can conclude that Algorithm 2 is numerically more trustworthy than Algorithm 1.

Results and Discussion

In what follows we run two algorithms on three sets of data. Also, we assume that the set of numbers is $\mathbb{F}=\mathbb{F}(10,4,L,U)$. Our aim is to calculate the relative errors of the two algorithms in the set $\mathbb{F}=\mathbb{F}(10,6,L,U)$.

Remark 1. *e_1 and e_2 denote the relative error for Algorithm 1 and 2, respectively*.

Example 1. *We assume that the set of observations is x_1=0.1256, x_2=0.2347, x_3=0.3511. In the set $\mathbb{F}=\mathbb{F}(10,4,L,U)$ we obtain $\bar{x}$ =0.2371 and in the set $\mathbb{F}=\mathbb{F}(10,6,L,U)$ we obtain $\bar{x}$ =0.237133. The variance and relative errors are shown n Table 1*.

Example 2. *We assume that the set of observations is x_1=0.1123, x_2=0.1234, x_3=0.1356. In the set $\mathbb{F}=\mathbb{F}(10,4,L,U)$ we obtain $\bar{x}$ =0.1238 and in the set $\mathbb{F}=\mathbb{F}(10,6,L,U)$ we obtain $\bar{x}$ =0.123767. Therefore, we have Table 2*:

Example 3. *We assume that the set of observations is x_1=0.1113, x_2=0.1124, x_3=0.1135. In the set $\mathbb{F}=\mathbb{F}(10,4,L,U)$ we obtain $\bar{x}$ =0.1124 and in the set $\mathbb{F}=\mathbb{F}(10,6,L,U)$ we obtain $\bar{x}$ =0.112400. Therefore, we have Table 3*:

We have proven that the total effect of rounding error of Algorithm 2 is less that the total effect of rounding error of Algorithm 1. Hence, we can conclude that Algorithm 2 is numerically more trustworthy. We should notice that for large n, computing $\sum_{i=1}^{n}x_i{}^2$ and $\sum_{i=1}^{n}(x_i-\bar{x})^2$ is a numerical problem and we should add small data first to prevent large errors.

Table 1. Comparison between two Algorithms for example 1

	Alg. 1	**Alg. 2**
Variance of observations in $\mathbb{F}$ (10,4,*L*,*U*)	$\bar{S}_1^2$=0.0128	$\bar{S}_2^2$=0.0127
Variance of observations in $\mathbb{F}$ (10,6,*L*,*U*)	S_1^2=0.012717	S_2^2=0.012717
Relative error in $\mathbb{F}$ (10,6,*L*,*U*)	e_1=0.006527	e_2=0.001337

Table 2. Comparison between two Algorithms for example 2

	Alg. 1	**Alg. 2**
Variance of observations in $\mathbb{F}$ (10,4,*L*,*U*)	$\bar{S}_1^2$=0.0002	$\bar{S}_2^2$=0.0001
Variance of observations in $\mathbb{F}$ (10,6,*L*,*U*)	S_1^2=0.000136	S_2^2=0.000136
Relative error in $\mathbb{F}$ (10,6,*L*,*U*)	e_1=0.470588	e_2=0.264706

Table 3. Comparison between two Algorithms for example 3

	Alg. 1	**Alg. 2**
Variance of observations in $\mathbb{F}$ (10,4,*L*,*U*)	$\bar{S}_1^2$=0.0001	$\bar{S}_2^2$=0.0000
Variance of observations in $\mathbb{F}$ (10,6,*L*,*U*)	S_1^2=0.000001	S_2^2=0.000001
Relative error in $\mathbb{F}$ (10,6,*L*,*U*)	e_1=0.990000×10^2	e_2=0.100000×10^1

Acknowledgement

The authors wish to thank the referees and the editor for their useful comments and suggestions.

References

1. R. L. Ashenhurst & N. Metropolis, Un-normalized floating-point arithmetic, J. Assoc. Comput. Math. **6:**415-428 (1959).
2. F. L. Bauer, Computational graphs and rounding error, SIAM J. Numer. Anal. **11:** 87-96 (1974).
3. F. L. Bauer, J. Heinhold, k. Samelson & R. Sauer, *Modern Rechenanlagen, Teubner, Sttgart*, (1965).
4. H. Henrici, *Error Propagation for Difference Methods*,

Wiley, New York, (1963).
5. D. E. Knuth, *The Art of Computer programming, Semi-numerical Algorithms*, Vol. 2, Addison-Wesley, Reading, MA,(1969).
6. U. Kulisch, Grundzgue der Intervallrechning,in: D. Lauugwitz (Ed.), Uberblicke Mathematik, Vol. 2, Bibliographihes Institut Mannheim, pp.51-98 (1969).
7. R. E. Moore, *Interval Analysis*, Prentice-Hal, Englewood Cliff, NJ, (1966).
8. J. Von Neumann & H. H. Goldstein, Numerical inverting of matrices, Bull. Amer. Math,Soc. **53**: 1021-1099 (1947).
9. H. A. Rademacher, On the accumulation of errors in processes of integration on the high-speed calculating machines, in: Proceedings of a Symposium on Large Scale Digital Calculating Machinery, Annals Comput. Labor. Harvard University, Vol. **16:** pp. 176-185 (1948).
10. M. Rosenbaum, Integrated volatility and round-off error. Bernoulli **15:** No 3, 687-720 :(2009).
11. D. Rostamy V. F., A Stochastic Partial Differential Equation for Computational Algorithms, **159:** 429-434 (2004).
12. J. b. Scarborough, *Numerical Mathematical Analysis*, second ed. , Johns Hopkins Press, Baltimore, MA,(2002).
13. O. Stauning, *Automatic Validation of Numerical Solutions*, Ph. D. Thesis, Technical University of Denmark, Lyngby, (1997).
14. F. H. Sterbenz, *Floating Point Computation*, Prentice-Hall, Englewood Cliffs, NJ, (1974).
15. J. Stoer & R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York,(2002).
16. J. H. Wilkinson, *Error analysis of floating-point Computation*, Numer. Math. 2(1960)219-340.
17. J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Wiley, New York, (1963).
18. J. H. Wilkinson, *The Algebraic Eigen-value Problem*, Clarendon Press, Oxford, (1965).