Comput Stat (2016) 31:1305–1325
DOI 10.1007/s00180-015-0637-z

ORIGINAL PAPER

Numerically stable, scalable formulas for parallel and online computation of higher-order multivariate central moments with arbitrary weights

Philippe Pébay[1] · Timothy B. Terriberry[2] ·
Hemanth Kolla[3] · Janine Bennett[4]

Received: 26 January 2015 / Accepted: 9 December 2015 / Published online: 29 March 2016

Abstract Formulas for incremental or parallel computation of second order central moments have long been known, and recent extensions of these formulas to univariate and multivariate moments of arbitrary order have been developed. Such formulas are of key importance in scenarios where incremental results are required and in parallel and distributed systems where communication costs are high. We survey these recent results, and improve them with arbitrary-order, numerically stable one-pass formulas which we further extend with weighted and compound variants. We also develop a generalized correction factor for standard two-pass algorithms that enables

Dedicated to the Memory of Dr Timothy J. Baker (1948–2006).

Philippe Pébay, Hemanth Kolla and Janine Bennett: These authors were supported by the United States Department of Energy, Office of Science, Office of Defense, and Sandia LDRD Program. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94-AL85000.

✉ Philippe Pébay
pppebay@sandia.gov

Timothy B. Terriberry
tterribe@xiph.org

Hemanth Kolla
hnkolla@sandia.gov

Janine Bennett
jcbenne@sandia.gov

[1] Sandia National Laboratories, MS 9159, P.O. Box 969, Livermore, CA 94551, USA

[2] The Xiph.Org Foundation, 2521 S. Oxford St., Arlington, VA 22206, USA

[3] Sandia National Laboratories, MS 9158, P.O. Box 969, Livermore, CA 94551, USA

[4] Sandia National Laboratories, MS 9152, P.O. Box 969, Livermore, CA 94551, USA

the maintenance of accuracy over nearly the full representable range of the input, avoiding the need for extended-precision arithmetic. We then empirically examine algorithm correctness for pairwise update formulas up to order four as well as condition number and relative error bounds for eight different central moment formulas, each up to degree six, to address the trade-offs between numerical accuracy and speed of the various algorithms. Finally, we demonstrate the use of the most elaborate among the above mentioned formulas, with the utilization of the compound moments for a practical large-scale scientific application.

Keywords Descriptive statistics · Statistical moments · Parallel computing · Large data analysis

1 Introduction

Central moments, including the variance, and derived quantities like skewness and kurtosis, are some of the most widely used tools in descriptive statistics. However, standard approaches for computing them, reviewed in Sect. 2, either require two passes over the data, or are grossly inaccurate for data that is not contained within a very limited range. This poses a problem in streaming settings where incremental results are needed after each new value is observed, and for very large datasets, which may not fit in available memory, and increasingly are distributed over a number of hosts. The prevalence of large, distributed data sets has lead to the recent development of new statistical packages to analyze them, cf. Wylie et al. (2008), Wong et al. (2008), Bennett et al. (2009), Pébay et al. (2010, 2011), Eddelbuettel (2010), Schmidberger et al. (2009), Stata (2010). In this setting the cost of distributed memory access is so large that two-pass algorithms become entirely impractical. Even a single machine increasingly performs large parallel computations on a Graphics Processing Unit (GPU), where memory bandwidth is a significant bottleneck. Using two passes doubles the execution time, and using double precision arithmetic doubles it again, almost irrespective of the number of arithmetic operations performed in each pass.

For the second central moment (the variance), accurate, one-pass, incremental approaches have long been known, e.g. Welford (1962), Neely (1966), West (1979). Chan et al. (1979) generalized them into a "pairwise algorithm", which computes the variance of a set by partitioning it into two subsets, computing their second order statistics recursively, and then combining them with an updating rule to obtain the second order statistics of the whole set. Constraining the second set to be a singleton yields an efficient incremental (on-line) algorithm. Alternatively, using subsets of roughly equal size yields a highly parallel algorithm. The latter also ensures intermediate terms will be commensurate, increasing accuracy by preventing destructive underflow. Incremental formulas for cumulants up to fourth order have been proposed for a zero-mean process, cf. Amblard and Brossier (1995), Dembélé and Favier (1998). Ensuring a zero-mean process involves removing a mean estimated from the data, which traditionally requires a two-pass algorithm, eliminating the benefit of a recursive update formula.

In Bennett et al. (2009), we generalized the variance calculation approach of Chan et al. (1979) to moments of arbitrary order, and formulas for incremental and pairwise algorithms were provided. These formulas are particularly useful as a number of applications of higher order moments require on-line updates or parallel processing. For instance, many communications applications use both univariate (Mendel 1991) and multivariate (Nikias and Mendel 1993) moments up to fourth order—or cumulants, which are frequently computed from the central moments. These include blind deconvolution (Shalvi and Weinstein 1990), blind source separation (Tugnait 1997), direction finding (Porat and Friedlander 1991), and speech detection (Nemer et al. 2001), all of which can benefit from on-line updates to adapt to changing channel conditions and minimize delay. Image processing also makes frequent use of higher-order moments for modeling non-linear distortions, with applications in deblurring (Xu and Crebbin 1996; Ibrahim et al. 1998; Wang et al. 2006), noise removal (Kleihorst et al. 1997), gamma correction and radial distortion estimation (Farid and Popescu 2001), and steganalysis (Lyu and Farid 2002). Skewness and kurtosis are also commonly used in financial modeling (Samuelson 1970; Harvey and Siddique 2000), where datasets are so large that distributed processing is required. Although examples are less common, moments up to sixth order can aid chromatic dispersion compensation in long distance fiber-optic lines (Kikuchi et al. 2005), and eighth-order moments provide a means to identify cell phone modulation schemes (Prakasam and Madheswaran 2008), to name a few. In this paper we further expand these formulas to a variety of other extensions: weighted moments, forgetting schemes, and compound moments. Compound moments have important applications for turbulent flow analysis (Jones 1993) and we demonstrate their application in this setting.

As the order of the moment increases, even the venerable two-pass algorithm may be inaccurate, as the numerical error for evaluating polynomials around the mean grows exponentially with the degree. When communication costs are the bottleneck, doubling the working precision doubles the computation time. Alternatives, such as compensation algorithms for summation (Ogita et al. 2005) and polynomial evaluation (Langlois and Louvet 2007), require twice as much storage for intermediate values. This is not an issue when computations are performed locally, but for distributed computations this is just as costly as doubling the working precision. A well-known correction factor, attributed by Chan et al. (1983) to Åke Björk, though also proposed by Neely (1966), greatly improves the accuracy of the two-pass algorithm when computing the variance.

In this paper we generalize this correction factor to moments of arbitrary order. Our scheme transmits only one additional value in the second pass, but can correct for the error in moments of all orders, providing increased accuracy for higher order moments at a fraction of the cost of generic compensation schemes. We provide numerical results for most of these new formulas, including comparisons with other formulations and an application to a scientific use case. In particular, we empirically observe that our generalized incremental and pairwise algorithms perform almost as well as the two-pass algorithm, and in some cases even better. Our corrected two-pass algorithm yields accurate results over almost the entire range of representable input values.

2 Background: computing statistical moments

We begin with a brief notational preamble, after which we directly formulate the main difficulties that arise when computing statistical moments using floating point representations.

2.1 Statement of the problem

For p a non-negative integer and using $E\left[\cdot\right]$ to denote the expectation, the pth central moment of a (univariate, real) random variable X is defined as

$$\mu_p \triangleq E\left[(X - E[X])^p\right], \tag{2.1}$$

when the expectations exist (some random variables, e.g. those with a Cauchy distribution, do not have an expectation). For a finite population of n equiprobable values in a multiset $S = \{x_i\}_{i=1}^{n}$, this reduces to

$$\mu_p = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^p \tag{2.2}$$

where

$$\bar{x} \triangleq \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.3}$$

is the mean. Note that S has to be a multiset, not a set, in order to allow for multiple copies of the same value.

The first central moment is exactly zero, and the second central moment is the *variance*, $\sigma^2 \triangleq \mu_2$. For this paper, we only consider the statistics of finite populations taken in their entirety, i.e., not sampled, to avoid issues of estimation bias. If S is instead just a finite sample of an infinite population, one may obtain unbiased estimates of the moments of the whole population (Halmos 1946). E.g., for $p = 2$ and $p = 3$ the unbiased estimators are $\frac{n}{n-1}\mu_2$ and $\frac{n^2}{(n-1)(n-2)}\mu_3$, respectively, with μ_p computed over the sample as in (2.2). For higher order the formulas become more complicated; for instance, with $p = 4$ the unbiased estimator is (Dodge and Rousson 1999):

$$\frac{(n-1)(n^2-3n+3)}{n^3}\mu_4 + \frac{3(2n-3)(n-1)}{n^3}{\mu_2}^2.$$

Samples drawn (without replacement) from a finite population require additional corrections. However, unbiased estimates of the moments do not, in general, lead to unbiased estimates of the derived quantities, such as standard deviation, skewness, and kurtosis.

2.2 One-pass versus two-pass approaches

The standard *two-pass algorithm* explicitly computes the mean μ_p using (2.3), followed by (2.2) to obtain the variance. The two-pass algorithm is numerically stable even when $\bar{x}$ is large and μ_p is small, in a sense that is made more precise in Sect. 4.2. Its stability for $p = 2$ can be further improved by applying a well-known correction factor (Neely 1966; Chan et al. 1983):

$$\mu_2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 - \mu_1{}^2 \tag{2.4}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i - \bar{x}\right)^2. \tag{2.5}$$

By definition, $\mu_1 = 0$ when evaluated with exact arithmetic, but Chan et al. (1983) show that when computing both μ_1 and μ_2 with inexact arithmetic using the two-pass algorithm, the rounding error introduced into μ_1 cancels much of the rounding error introduced into μ_2. The *corrected two-pass algorithm* still only requires two passes, since μ_1 and μ_2 can be computed simultaneously.

However, the two-pass approach is inadequate for large or distributed data sets, where making two complete passes through the data is extremely expensive. It is also unsuitable when one needs a new estimate of μ_p each time a new x value is obtained.

The obvious method of obtaining a one-pass calculation, what Chan et al. (1979) call the *textbook algorithm* for the variance, is to expand the product $(x - \bar{x})^p$ into explicit powers of x and $\bar{x}$. Using the binomial theorem, this is easy to generalize to arbitrary order:

$$\mu_p = \sum_{k=0}^{p}\binom{p}{k}\left(\frac{1}{n}\sum_{i=1}^{n} x_i^{p-k}\right)(-\bar{x})^k. \tag{2.6}$$

The inner sums, including that for $\bar{x}$, can be updated incrementally or computed in parallel, and the outer sum requires negligible additional work, since p is typically small. However, even for $p = 2$, this expression quickly becomes grossly inaccurate, as our experiments in Sect. 4.2 show. The alternating signs on each term cause destructive cancellation, and few, if any, significant digits are retained. The results may even be negative when p is even, which is clearly nonsensical for it violates the Cauchy-Schwarz inequality.

Example 2.1 Consider the values $x_1 = 1$ and $x_2 = x_3 = x_4 = 1 + 10^{-13}$. Their respective double-precision (64 bits) IEEE-754 floating point hexadecimal representations (IEEE 1985) are

$$r(x_1) = \texttt{3ff0000000000000}$$

and

$$r(x_2) = r(x_3) = r(x_4) = \texttt{3ff00000000001c2}.$$

Subsequently, one obtains the following representations for the mean:

$$r(\mu) = r\left(\frac{1}{4}\sum_{i=1}^{4} r(x_i)\right) = \texttt{3ff0000000000152},$$

and the mean of the squares:

$$r\left(\frac{1}{4}\sum_{i=1}^{4} r(x_i)^2\right) = \texttt{3ff00000000002a3}.$$

Thus the textbook algorithm yields the following value for the variance:

$$r\left(\frac{1}{4}\sum_{i=1}^{4} r(x_i)^2\right) - r(\mu)^2 = \texttt{bcb0000000000000},$$

which represents a negative number:

$$r(-2.220446049250313 \times 10^{-16}) = \texttt{bcb0000000000000}$$

thus establishing that the textbook algorithm can yield negative variances even with small data sets. This problem is therefore not limited to large statistical calculations, but it becomes potentially worse as the size of the sample set increases.

2.3 Numerically stable one-pass algorithms

Much better one-pass algorithms for computing the variance have long been known (Welford 1962; Neely 1966; West 1979; Chan et al. 1979); Chan et al. (1979) summarize them using a generic set of recurrence formulas. Partition S into multisets $\mathcal{A}$ and $\mathcal{B}$ of respective sizes $n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ and define $\mu_{p,\mathcal{A}}$, $\mu_{p,\mathcal{B}}$, $\bar{x}_{\mathcal{A}}$, and $\bar{x}_{\mathcal{B}}$ to be the corresponding statistic computed over each partition. Then let

$$\delta_{\mathcal{B},\mathcal{A}} \triangleq \bar{x}_{\mathcal{B}} - \bar{x}_{\mathcal{A}}, \tag{2.7}$$

$$M_p \triangleq n\mu_p, \tag{2.8}$$

and again give $M_p^{\mathcal{A}}$ and $M_p^{\mathcal{B}}$ an equivalent definition restricted to each partition. We will find it more convenient to work with these M_p quantities, rather than μ_p, though either may be readily obtained from the other. Now

$$\bar{x} = \bar{x}_{\mathcal{A}} + \frac{n_{\mathcal{B}}}{n}\delta_{\mathcal{B},\mathcal{A}}, \tag{2.9}$$

$$M_2 = M_2^{\mathcal{A}} + M_2^{\mathcal{B}} + \frac{n_{\mathcal{A}} n_{\mathcal{B}}}{n} \delta_{\mathcal{B},\mathcal{A}}^2 . \tag{2.10}$$

A number of algorithms can be derived from these simple recurrences. Letting $n_{\mathcal{B}} = 1$ so that $\mathcal{B} = \{y\}$ a singleton yields the incremental update formulas of West (1979):

$$\bar{x} = \bar{x}_{\mathcal{A}} + \frac{y - \bar{x}_{\mathcal{A}}}{n}, \tag{2.11}$$

$$M_2 = M_2^{\mathcal{A}} + \frac{n-1}{n} (y - \bar{x}_{\mathcal{A}})^2 . \tag{2.12}$$

On the other hand, letting $n_{\mathcal{A}} = n_{\mathcal{B}} = n/2$ (assuming n is even) gives a recursive pairwise algorithm:

$$\bar{x} = \bar{x}_{\mathcal{A}} + \frac{1}{2} \delta_{\mathcal{B},\mathcal{A}}, \tag{2.13}$$

$$M_2 = M_2^{\mathcal{A}} + M_2^{\mathcal{B}} + \frac{n}{4} \delta_{\mathcal{B},\mathcal{A}}^2 . \tag{2.14}$$

While easily parallelizable, the pairwise algorithm can also reduce destructive underflow on a uniprocessor, since it ensures that the terms in the update formulas are approximately commensurate when the data is independent and identically distributed. By contrast, when n is large, the terms corresponding to the $\mathcal{B}$ partition in (2.11) and (2.12) are very small, affecting only a few of the least significant digits of $\bar{x}$ and M_2. Both algorithms perform the same number of updates, but the pairwise algorithm requires $O(\log n)$ additional storage. The same pairwise strategy can be applied with similar benefits to *all* summation formulas, including (2.2), (2.3), and (2.6).

3 Arbitrary-order update formulas

3.1 Univariate formulas

We begin by generalizing all pairwise and update formulas to arbitrary order and arbitrary set decomposition.

Proposition 3.1 *For any integer $p \geq 2$,*

$$\begin{aligned} M_p = {} & M_p^{\mathcal{A}} + M_p^{\mathcal{B}} + n_{\mathcal{A}} \left(\frac{-n_{\mathcal{B}}}{n} \delta_{\mathcal{B},\mathcal{A}} \right)^p + n_{\mathcal{B}} \left(\frac{n_{\mathcal{A}}}{n} \delta_{\mathcal{B},\mathcal{A}} \right)^p \\ & + \sum_{k=1}^{p-2} \binom{p}{k} \delta_{\mathcal{B},\mathcal{A}}^k \left[M_{p-k}^{\mathcal{A}} \left(\frac{-n_{\mathcal{B}}}{n} \right)^k + M_{p-k}^{\mathcal{B}} \left(\frac{n_{\mathcal{A}}}{n} \right)^k \right]. \end{aligned} \tag{3.1}$$

Proof By the definition of M_p, and because $\{\mathcal{A}, \mathcal{B}\}$ is a partition of S, one has

$$
\begin{aligned}
M_p &= \sum_{i=1}^{n} (x_i - \bar{x})^p && (3.2)\\
&= \sum_{i=1}^{n_\mathcal{A}} (x_i - \bar{x})^p + \sum_{i=n_\mathcal{A}+1}^{n} (x_i - \bar{x})^p && (3.3)\\
&= \sum_{i=1}^{n_\mathcal{A}} \left(x_i - \frac{n_\mathcal{A}\bar{x}_\mathcal{A} + n_\mathcal{B}\bar{x}_\mathcal{B}}{n} \right)^p + \sum_{i=n_\mathcal{A}+1}^{n} \left(x_i - \frac{n_\mathcal{A}\bar{x}_\mathcal{A} + n_\mathcal{B}\bar{x}_\mathcal{B}}{n} \right)^p && (3.4)\\
&= \sum_{i=1}^{n_\mathcal{A}} \left(x_i - \bar{x}_\mathcal{A} - \frac{n_\mathcal{B}}{n}\delta_{\mathcal{B},\mathcal{A}} \right)^p + \sum_{i=n_\mathcal{A}+1}^{n} \left(x_i - \bar{x}_\mathcal{B} + \frac{n_\mathcal{A}}{n}\delta_{\mathcal{B},\mathcal{A}} \right)^p && (3.5)\\
&= \sum_{k=0}^{p} \binom{p}{k} \left(\frac{\delta_{\mathcal{B},\mathcal{A}}}{n} \right)^k \left[(-1)^k M_{p-k}^{\mathcal{A}} n_\mathcal{B}^k + M_{p-k}^{\mathcal{B}} n_\mathcal{A}^k \right], && (3.6)
\end{aligned}
$$

thanks to the commutativity of summation over finite sets, which allows us to swap $\sum_{k=0}^{p}$ with $\sum_{i=1}^{n_\mathcal{A}}$ and $\sum_{i=n_\mathcal{A}+1}^{n}$. Now, a few simplifications are in order: first, the $k = 0$ term of the above summation is simply $M_p^\mathcal{A} + M_p^\mathcal{B}$; second, by definition, both $M_1^\mathcal{A}$ and $M_1^\mathcal{B}$ are zero, eliminating the $k = p-1$ term; last, $M_0^\mathcal{A} = n_\mathcal{A}$ and $M_0^\mathcal{B} = n_\mathcal{B}$, eliminating the need to compute these values separately for use in the $k = p$ term. Applying these three simplifications to (3.6) yields (3.1). □

The computation of M_p requires the values of $M_q^\mathcal{A}$ and $M_q^\mathcal{B}$ for each $q \in \{2, \dots, p\}$, instead of just for $q = p$. Thus the update formula performs $O(p^2)$ arithmetic operations per element, compared to the $O(1)$ operations the two-pass algorithm required if only M_p is actually needed. A small improvement in accuracy may be obtained by evaluating (3.1) as the sum of two polynomials in $\frac{-n_\mathcal{B}}{n}\delta_{\mathcal{B},\mathcal{A}}$ and $\frac{n_\mathcal{A}}{n}\delta_{\mathcal{B},\mathcal{A}}$ using Horner's rule. When local computations are cheap, one could even use the compensated Horner scheme, which often provides exactly rounded results (Langlois and Louvet 2007), but this does not prevent the accumulation of rounding errors in recursive applications of (3.1).

Corollary 3.1 *In the case where $\mathcal{B}$ is reduced to a singleton $\{y\}$, Proposition 3.1 reduces to the incremental update formula for $S = \mathcal{A} \cup \{y\}$ as follows*

$$
M_p = M_p^\mathcal{A} + \left[\frac{n-1}{(-n)^p} + \left(\frac{n-1}{n} \right)^p \right] \delta_{\mathcal{B},\mathcal{A}}^p + \sum_{k=1}^{p-2} \binom{p}{k} M_{p-k}^\mathcal{A} \left(\frac{-\delta_{\mathcal{B},\mathcal{A}}}{n} \right)^k. \quad (3.7)
$$

Proof Corollary 3.1 is an immediate specialization of Proposition 3.1 obtained when $n_\mathcal{A} = n - 1$ and $n_\mathcal{B} = 1$. In this case, each $M_p^\mathcal{B}$ vanishes since $\bar{x}_\mathcal{B} = y$, and thus (3.1) immediately simplifies to (3.7). □

Remark 3.1 By noticing that

$$\frac{n-1}{(-n)^2}+\left(\frac{n-1}{n}\right)^2=\frac{n^2-n}{n^2}=\frac{n-1}{n} \tag{3.8}$$

and taking $p = 2$, one directly retrieves (2.12) from Corollary 3.1.

We provide implementations of univariate incremental and pairwise update formulas in the descriptive statistics class `vtkDescriptiveStatistics` of the open-source Visualization Tool Kit (VTK), with the `Learn()` and `Aggregate()` methods, respectively.

3.2 Multivariate formulas

We continue by generalizing the univariate results to arbitrary multivariate moments (co-moments). These are of interest, in particular, for Pearson correlation analysis, which we wish to conduct on large-scale, distributed data sets. Higher order co-moments such as co-skewness and co-kurtosis also have financial modeling applications (Hung et al. 2004).

Extending the notation of previous sections, let $S = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, where each $x_i = (x_{i,1}, \dots, x_{i,d}) \in \mathbb{R}^d$ is a d-dimensional vector. Now let $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$ and $\beta = (\beta_1, \dots, \beta_d) \in \mathbb{N}^d$ be *multi-indices* of non-negative integers so that

$$\alpha \le \beta \iff \alpha_j \le \beta_j \ \forall j \in \{1 \dots d\}, \tag{3.9}$$

$$|\alpha| \triangleq \sum_{j=1}^{d} \alpha_j, \tag{3.10}$$

$$\binom{\alpha}{\beta} \triangleq \prod_{j=1}^{d} \binom{\alpha_j}{\beta_j}, \tag{3.11}$$

$$x_i^\alpha \triangleq \prod_{j=1}^{d} x_{i,j}^{\alpha_j}, \tag{3.12}$$

and define the central co-moment of order α of a finite population S to be

$$\mu_\alpha \triangleq \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^\alpha. \tag{3.13}$$

Under this definition the usual covariance is obtained with $\alpha = (1, 1)$. All the terms $\bar{x}$, $\bar{x}_{\mathcal{A}}$, $\bar{x}_{\mathcal{B}}$, $\delta_{\mathcal{B},\mathcal{A}}$, $\mu_{\alpha,\mathcal{A}}$, $\mu_{\alpha,\mathcal{B}}$, M_α, $M_\alpha^{\mathcal{A}}$, $M_\alpha^{\mathcal{B}}$ are defined exactly as in the univariate case, with α replacing the univariate order p. Similarly, we define

$$M_\alpha \triangleq \sum_{i=1}^{n}(x_i-\bar{x})^\alpha \tag{3.14}$$

for the multi-index α.

Proposition 3.2 *The recursive update formula for M_α is:*

$$M_\alpha = \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} \delta_{\mathcal{B},\mathcal{A}}^{\beta} \left[\left(-\frac{n_\mathcal{B}}{n} \right)^{|\beta|} M_{\alpha-\beta}^{\mathcal{A}} + \left(\frac{n_\mathcal{A}}{n} \right)^{|\beta|} M_{\alpha-\beta}^{\mathcal{B}} \right]. \tag{3.15}$$

Proof Following the proof of Proposition 3.1,

$$M_\alpha = \sum_{i=1}^{n_\mathcal{A}} \left(x_i - \bar{x}_\mathcal{A} - \frac{n_\mathcal{B}}{n} \delta_{\mathcal{B},\mathcal{A}} \right)^\alpha + \sum_{i=n_\mathcal{A}+1}^{n} \left(x_i - \bar{x}_\mathcal{B} + \frac{n_\mathcal{A}}{n} \delta_{\mathcal{B},\mathcal{A}} \right)^\alpha. \tag{3.16}$$

Expanding out the multi-index products and applying the binomial theorem yields

$$\begin{aligned} M_\alpha = & \sum_{i=1}^{n_\mathcal{A}} \prod_{j=1}^{d} \sum_{k=0}^{\alpha_j} \binom{\alpha_j}{k} \left(x_{i,j} - \bar{x}_{j,\mathcal{A}} \right)^{\alpha_j - k} \left(-\frac{n_\mathcal{B}}{n} \delta_{\mathcal{B},\mathcal{A}_j} \right)^k \\ & + \sum_{i=n_\mathcal{A}+1}^{n} \prod_{j=1}^{d} \sum_{k=0}^{\alpha_j} \binom{\alpha_j}{k} \left(x_{i,j} - \bar{x}_{j,\mathcal{B}} \right)^{\alpha_j - k} \left(\frac{n_\mathcal{A}}{n} \delta_{\mathcal{B},\mathcal{A}_j} \right)^k. \end{aligned} \tag{3.17}$$

Distributing the inner sums over the products, this simplifies to

$$\begin{aligned} M_\alpha = & \sum_{i=1}^{n_\mathcal{A}} \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} \left(x_i - \bar{x}_\mathcal{A} \right)^{\alpha-\beta} \left(-\frac{n_\mathcal{B}}{n} \delta_{\mathcal{B},\mathcal{A}} \right)^\beta \\ & + \sum_{i=n_\mathcal{A}+1}^{n} \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} \left(x_i - \bar{x}_\mathcal{B} \right)^{\alpha-\beta} \left(\frac{n_\mathcal{A}}{n} \delta_{\mathcal{B},\mathcal{A}} \right)^\beta. \end{aligned} \tag{3.18}$$

Once again, the commutativity of summation over finite sets allows us to swap $\sum_{\beta \leq \alpha}$ with $\sum_{i=1}^{n_\mathcal{A}}$ and $\sum_{i=n_\mathcal{A}+1}^{n}$, and rearranging terms produces (3.15). □

A quick check verifies that (3.15) reduces to (3.6) when $d = 1$ and $\alpha = (p)$. The same simplifications made in the univariate case still apply when actually computing the full (3.15), though they do not simplify the notation. That is, the $\beta = \mathbf{0}_d$ term is simply $M_\alpha^\mathcal{A} + M_\alpha^\mathcal{B}$, and for every β term where it exists $k \in \{1, \ldots, d\}$ such that $\alpha_k - \beta_k = 1$, one has

$$M_{\alpha-\beta} = \sum_{i=1}^{n} \prod_{j=1}^{d} (x_{i,j} - \bar{x}_j)^{\alpha_j - \beta_j} = \sum_{i=1}^{n} (x_{i,k} - \bar{x}_k) \prod_{\substack{j=1 \\ j \neq k}}^{d} (x_{i,j} - \bar{x}_j)^{\alpha_j - \beta_j} = 0 \tag{3.19}$$

by definition of the d-dimensional mean $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_d)$, and thus all such β terms vanish. Also, when $\beta = \alpha$, $M_\mathbf{0}^\mathcal{A} = n_\mathcal{A}$ and $M_\mathbf{0}^\mathcal{B} = n_\mathcal{B}$. Applying these simplifications to (3.15) with $\alpha = (1, 1)$ yields

$$M_{(1,1)} = M^{\mathcal{A}}_{(1,1)} + M^{\mathcal{B}}_{(1,1)} + \left[n_{\mathcal{A}} \left(-\frac{n_{\mathcal{B}}}{n} \right)^2 + n_{\mathcal{B}} \left(\frac{n_{\mathcal{A}}}{n} \right)^2 \right] \delta^{(1,1)}_{\mathcal{B},\mathcal{A}} \tag{3.20}$$

$$= M^{\mathcal{A}}_{(1,1)} + M^{\mathcal{B}}_{(1,1)} + \frac{n_{\mathcal{A}} n_{\mathcal{B}}}{n} (\bar{x}_{1,\mathcal{B}} - \bar{x}_{1,\mathcal{A}})(\bar{x}_{2,\mathcal{B}} - \bar{x}_{2,\mathcal{A}}). \tag{3.21}$$

When $\mathcal{B}$ is reduced to a singleton $\{(y_1, y_2)\}$, this is equivalent to the incremental covariance update formula derived by Neely (1966):

$$M_{(1,1)} = M^{\mathcal{A}}_{(1,1)} + M^{\mathcal{B}}_{(1,1)} + \frac{n-1}{n} (y_1 - \bar{x}_{1,\mathcal{A}})(y_2 - \bar{x}_{2,\mathcal{A}}). \tag{3.22}$$

We provide implementations of bivariate incremental and pairwise update formulas in the open-source Visualization Tool Kit (VTK), respectively in the `Learn()` and `Aggregate()` methods of the correlative statistics `vtkCorrelativeStatistics` class.

3.3 Weighted formulas

Consider a quantity $\widetilde{x}_W$ defined as a weighted arithmetic mean with respect to a given set of weights W. Replacing in Propositions 3.1 and 3.2 the set sizes n, $n_{\mathcal{A}}$, and $n_{\mathcal{B}}$ with sums of non-negative weights $\{w_i\}_{1 \leq i \leq N}$,

$$W \triangleq \sum_{i=1}^{n} w_i, \quad W_{\mathcal{A}} \triangleq \sum_{i=1}^{n_{\mathcal{A}}} w_i, \quad W_{\mathcal{B}} \triangleq \sum_{i=n_{\mathcal{A}}+1}^{n} w_i, \tag{3.23}$$

and the other sums with weighted sums,

$$\widetilde{x}_W \triangleq \frac{1}{W} \sum_{i=1}^{n} w_i x_i, \tag{3.24}$$

$$M_p \triangleq \sum_{i=1}^{n} w_i (x_i - \widetilde{x}_W)^p, \tag{3.25}$$

leads directly to weighted versions of our main results:

$$\widetilde{x}_W = \widetilde{x}_{W,\mathcal{A}} + \frac{W_{\mathcal{B}}}{W} \delta_{\mathcal{B},\mathcal{A}}, \tag{3.26}$$

$$M_p = M^{\mathcal{A}}_p + M^{\mathcal{B}}_p + W_{\mathcal{A}} \left(\frac{-W_{\mathcal{B}}}{W} \delta_{\mathcal{B},\mathcal{A}} \right)^p + W_{\mathcal{B}} \left(\frac{W_{\mathcal{A}}}{W} \delta_{\mathcal{B},\mathcal{A}} \right)^p$$

$$+ \sum_{k=1}^{p-2} \binom{p}{k} \delta^k_{\mathcal{B},\mathcal{A}} \left[M^{\mathcal{A}}_{p-k} \left(\frac{-W_{\mathcal{B}}}{W} \right)^k + M^{\mathcal{B}}_{p-k} \left(\frac{W_{\mathcal{A}}}{W} \right)^k \right], \tag{3.27}$$

$$M_\alpha = \sum_{\beta \le \alpha} \binom{\alpha}{\beta} \delta_{\mathcal{B},\mathcal{A}}^{\beta} \left[M_{\alpha-\beta}^{\mathcal{A}} \left(-\frac{W_\mathcal{B}}{W} \right)^{|\beta|} + M_{\alpha-\beta}^{\mathcal{B}} \left(\frac{W_\mathcal{A}}{W} \right)^{|\beta|} \right]. \tag{3.28}$$

These formulas may be used to derive adaptive estimators for non-stationary signals by setting $W_\mathcal{B} = \frac{1}{\eta} W_\mathcal{A}$, where $\frac{1}{\eta}$ with $\eta > 0$ is a *forgetting factor*, similar to that proposed by Dembélé and Favier (1998). This holds the relative importance of the most recent sample constant, while that of past samples decays exponentially. Other adaptive schemes are possible.

Remark 3.2 In addition, the proofs of Propositions 3.1 and 3.2 remain equally valid if the sums over S are replaced with integrals, since the other sums are finite. Thus, one can use (3.27) and (3.28) to compute moments of mixture distributions given the moments of each independent mixture element. Such moments can indicate goodness of fit or even be used to estimate the mixture parameters themselves via the method of moments (Pearson 1894).

3.4 Formulas for compound moments

A special case is that of compound moments of the type

$$\mu_p = \frac{1}{n} \sum_{i=1}^{n} (x_i - \widetilde{x}_W)^p, \tag{3.29}$$

where $\widetilde{x}_W$ is the weighted mean defined by (3.24). Such compound moments are often used in moment-closure modelling methods of turbulent flows (Jones 1993). Correspondingly, we define the quantity $\widetilde{M}_p$ by replacing $\bar{x}$ with $\widetilde{x}_W$ in (3.2) and, when (3.26) is applied, it expands to

$$\widetilde{M}_p = \sum_{i=1}^{n_\mathcal{A}} (x_i - \widetilde{x}_W)^p + \sum_{i=n_\mathcal{A}+1}^{n} (x_i - \widetilde{x}_W)^p \tag{3.30}$$

$$= \sum_{i=1}^{n_\mathcal{A}} \left(x_i - \widetilde{x}_{W,\mathcal{A}} - \frac{W_\mathcal{B}}{W} \delta_{\mathcal{B},\mathcal{A}} \right)^p + \sum_{i=n_\mathcal{A}+1}^{n} \left(x_i - \widetilde{x}_{W,\mathcal{B}} + \frac{W_\mathcal{A}}{W} \delta_{\mathcal{B},\mathcal{A}} \right)^p \tag{3.31}$$

$$= \sum_{k=0}^{p} \binom{p}{k} \delta_{\mathcal{B},\mathcal{A}}^{k} \left[\widetilde{M}_{p-k}^{\mathcal{A}} \left(\frac{-W_\mathcal{B}}{W} \right)^k + \widetilde{M}_{p-k}^{\mathcal{B}} \left(\frac{W_\mathcal{A}}{W} \right)^k \right] \tag{3.32}$$

where $\widetilde{\delta}_{\mathcal{B},\mathcal{A}}$ is the compound counterpart of $\delta_{\mathcal{B},\mathcal{A}}$, defined as

$$\widetilde{\delta}_{\mathcal{B},\mathcal{A}} \triangleq \widetilde{x}_{W,\mathcal{B}} - \widetilde{x}_{W,\mathcal{A}}. \tag{3.33}$$

The $k = 0$ term simplifies to $\widetilde{M}_p^{\mathcal{A}} + \widetilde{M}_p^{\mathcal{B}}$, while the $k = p$ term assumes the slightly different form of

$$n_\mathcal{A} \left(-\frac{W_\mathcal{B}}{W} \widetilde{\delta}_{\mathcal{B},\mathcal{A}} \right)^p + n_\mathcal{B} \left(\frac{W_\mathcal{A}}{W} \widetilde{\delta}_{\mathcal{B},\mathcal{A}} \right)^p, \tag{3.34}$$

since $\widetilde{M}_0^{\mathcal{A}} = n_{\mathcal{A}}$ and $\widetilde{M}_0^{\mathcal{B}} = n_{\mathcal{B}}$. However, the $k = p - 1$ term is non-zero and expands to

$$p\widetilde{M}_1^{\mathcal{A}}\Big(-\frac{W_{\mathcal{B}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p-1} + p\widetilde{M}_1^{\mathcal{B}}\Big(\frac{W_{\mathcal{A}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p-1}, \tag{3.35}$$

where

$$\widetilde{M}_1^{\mathcal{A}} = n_{\mathcal{A}}\left(\bar{x}_{\mathcal{A}} - \widetilde{x}_{W,\mathcal{A}}\right) \tag{3.36}$$

and

$$\widetilde{M}_1^{\mathcal{B}} = n_{\mathcal{B}}\left(\bar{x}_{\mathcal{B}} - \widetilde{x}_{W,\mathcal{B}}\right). \tag{3.37}$$

The resulting expansion is thus

$$\begin{aligned}\widetilde{M}_p &= \widetilde{M}_p^{\mathcal{A}} + \widetilde{M}_p^{\mathcal{B}}\\ &\quad + p\widetilde{M}_1^{\mathcal{A}}\Big(\frac{-W_{\mathcal{B}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p-1} + p\widetilde{M}_1^{\mathcal{B}}\Big(\frac{W_{\mathcal{A}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p-1}\\ &\quad + n_{\mathcal{A}}\Big(\frac{-W_{\mathcal{B}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p} + n_{\mathcal{B}}\Big(\frac{W_{\mathcal{A}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p}\\ &\quad + \sum_{k=1}^{p-2}\binom{p}{k}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}^{k}\Big[\widetilde{M}_{p-k}^{\mathcal{A}}\Big(\frac{-W_{\mathcal{B}}}{W}\Big)^{k} + \widetilde{M}_{p-k}^{\mathcal{B}}\Big(\frac{W_{\mathcal{A}}}{W}\Big)^{k}\Big]\end{aligned} \tag{3.38}$$

which can be slightly simplified as follows

$$\begin{aligned}\widetilde{M}_p &= \widetilde{M}_p^{\mathcal{A}} + \widetilde{M}_p^{\mathcal{B}}\\ &\quad + \Big(p\widetilde{M}_1^{\mathcal{A}} - n_{\mathcal{A}}\frac{W_{\mathcal{B}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)\Big(\frac{-W_{\mathcal{B}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p-1}\\ &\quad + \Big(p\widetilde{M}_1^{\mathcal{B}} + n_{\mathcal{B}}\frac{W_{\mathcal{A}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)\Big(\frac{-W_{\mathcal{A}}}{W}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}\Big)^{p-1}\\ &\quad + \sum_{k=1}^{p-2}\binom{p}{k}\widetilde{\delta}_{\mathcal{B},\mathcal{A}}^{k}\Big[\widetilde{M}_{p-k}^{\mathcal{A}}\Big(\frac{-W_{\mathcal{B}}}{W}\Big)^{k} + \widetilde{M}_{p-k}^{\mathcal{B}}\Big(\frac{W_{\mathcal{A}}}{W}\Big)^{k}\Big].\end{aligned} \tag{3.39}$$

In Sect. 4.3 we apply this compound moment formula to a real scientific use case of turbulent combustion simulation data.

3.5 Two-pass correction

Chan et al. (1983) note that applying the correction factor for the second-order moment in (2.5) is equivalent to guessing a trial mean with (2.3), shifting the data by this value, and then applying the textbook algorithm.

Applying the same strategy to the arbitrary-order formula (2.6) yields

$$\mu_p = \sum_{k=0}^{p}\binom{p}{k}\Big(\sum_{i=1}^{n}(x_i - \bar{x})^{p-k}\Big)\Big(-\frac{1}{n}\sum_{i=1}^{n}x_i - \bar{x}\Big)^{k}. \tag{3.40}$$

Since central moments are defined relative to the mean, subtracting a constant from the data does not affect the result, but when that constant is the mean (or a close approximation), it can have a large effect on accuracy. Like the $p = 2$ case, there is no destructive cancellation since the correction terms are much smaller than the $k = 0$ term.

Again, this is a two-pass algorithm. The first pass computes $\bar{x}$, and the second pass computes all of the inner sums in parallel. It is inexpensive computationally, requiring only two additions per value plus a small, constant amount of work at the end. However, its real advantage over typical compensation schemes is that it can correct for the error in the moments of all orders while only transmitting a single correction term.

4 Numerical results

In this section, we examine the performance of the proposed formulas, in terms of numerical error and computational efficiency.

4.1 Algorithm correctness

We use a series of (pseudo-)randomly generated samples in order to experimentally assess the correctness of the algorithms obtained with the one-pass formulas of Sect. 3. Specifically, input tables are created at run time by generating 4 separate samples of independent pseudo-random variables, the two first (resp. last) variables having a standard normal (resp. standard uniform) distribution. For the sake of illustration, equally-sized subsets of data are created and their respective descriptive statistics calculated by each process using the incremental, one-pass formulas (3.7) and (3.21), respectively for descriptive and correlative statistics. Subsequentely, the aggregated statistics over the entire parallel data set are calculated using the pairwise update formulas (3.1) and (3.15), respectively.

Relatively large input sets are used ($n = 10^6$), in order to mitigate the risk of statistical bias due to insufficient sampling. In addition, the test case is run 100 times for each random variable, and we examine the statistical dispersion of the results of the ensemble of these runs. We compare the calculated results to the theoretical values of the random variables which serve as models for the pseudo-random inputs, namely, uniform over [0, 1] (denoted $\mathcal{U}(0, 1)$) and standard normal (denoted $\mathcal{N}(0, 1)$). This comparison is done by comparing the sample mean of the quantity of interest (e.g., mean) across a number n_r of runs to the corresponding theoretical quantity (e.g, expectation), and then by examining the variability of the results by checking the standard deviation of the quantity of interest across the n_r runs.

Using this methodology with either $n_r = 100$ or $n_r = 200$ runs over 32 processors, the results provided in Tables 1 and 2, respectively for the descriptive and correlative statistics pairwise update algorithms operating on standard uniform and standard normal pseudo-random inputs, we see that the numerical results agree with their theoretical counterparts, and display very limited variability across runs.

Table 1 Computed descriptive statistics of a pseudo-random sample (size: 10^6), averaged across 200 runs, versus theoretical values

Statistic	Exact	Sample mean	SD
Mean	0.5	0.4999973	5.968778×10^{-5}
Variance	$0.08\underline{3}\ldots$	8.333224×10^{-2}	1.292998×10^{-5}
Skewness	0	g_1 : -4.5×10^{-6} G_1 : -4.5×10^{-6}	g_1 : 3.0×10^{-4} G_1 : 3.0×10^{-4}
Kurtosis	-1.2	g_2 : -1.200003 G_2 : -1.200003	g_2 : 1.9×10^{-4} G_2 : 1.9×10^{-4}
Mean	0	-1.279931×10^{-6}	1.706985×10^{-4}
Variance	1	0.9999982	2.711815×10^{-4}
Skewness	0	g_1 : 1.5×10^{-4} G_1 : 1.5×10^{-4}	g_1 : 3.9×10^{-4} G_1 : 3.9×10^{-4}
Kurtosis	0	g_2 : 4.4×10^{-4} G_2 : 4.4×10^{-4}	g_2 : 8.9×10^{-4} G_2 : 8.9×10^{-4}

Top: $\mathcal{U}(0, 1)$, bottom: $\mathcal{N}(0, 1)$

Table 2 Computed correlative statistics of a pseudo-random sample (size: 10^6), averaged across 100 runs, versus theoretical values

Statistic	Exact	Sample mean	SD
Mean X	0.5	0.4999953	5.779502×10^{-5}
Mean Y	0.5	0.4999987	6.061554×10^{-5}
Variance X	$0.08\underline{3}\ldots$	8.33×10^{-2}	1.242125×10^{-5}
Variance Y	$0.08\underline{3}\ldots$	8.33×10^{-2}	1.267820×10^{-5}
Covariance	0	6.78×10^{-7}	1.622636×10^{-5}
Mean X	0	-6.86×10^{-6}	1.693682×10^{-4}
Mean Y	0	-2.71×10^{-7}	1.695545×10^{-4}
Variance X	1	1.000016	2.883309×10^{-4}
Variance Y	1	0.9999838	2.551778×10^{-4}
Covariance	0	-1.01×10^{-5}	1.885743×10^{-4}

Top: $\mathcal{U}(0, 1)$, bottom: $\mathcal{N}(0, 1)$

4.2 Condition number and relative error bounds

In order to examine the numerical stability of algorithms for computing the variance, Chan and Lewis (1978) introduced the concept of *condition number of the data* S, defined as follows

$$\kappa \triangleq \frac{\|S\|_2}{\sqrt{M_2}},$$

Table 3 Error bounds for the relative error in the calculation of M_2 as summarized in Chan et al. (1983)

Algorithm	Error bound
Textbook	$\kappa^2 \varepsilon N$
Pairwise textbook	$\kappa^2 \varepsilon \log N$
Two-pass	$\varepsilon N + \kappa^2 \varepsilon^2 N^2$
Pairwise two-pass	$\varepsilon \log N + \kappa^2 \varepsilon^2 \log^2 N$
Incremental	$\kappa \varepsilon N$
Pairwise	$\kappa \varepsilon \log N$ (conjecture)
Corrected two-pass	$\varepsilon N + \kappa^2 \varepsilon^3 N^3$
Pairwise corrected two-pass	$\varepsilon \log N + \kappa^2 \varepsilon^3 \log^3 N$

where $\|S\|_2$ is the Euclidean norm of the data set S. Therefore, using the fact that the mean of S is the minimizer of $u \mapsto \|S - u\|_2$, it immediately follows that $\kappa \geq 1$. The condition number quantifies the sensitivity of M_2 to relative errors introduced in the data S: if relative errors of size δ are introduced in S, then the resulting relative variation of M_2 is bounded by $\kappa\delta$.

Further, Chan et al. (1983) established error bounds for a variety of algorithms for computing variance, one and two-pass alike, that depend solely on κ, N, and ε. For convenience, we reproduce *in extenso* in Table 3 the error bounds for the relative error in the calculation of M_2 summarized in Chan et al. (1983); note that only the dominant terms are provided. Constant factors are omitted for clarity.

We empirically extended these results by comparing all 8 schemes present in the second-order analysis of Chan et al. (1983), up to the order 6. These results, averaged over 2^{16} Gaussian (resp. exponential) samples of various sizes, are illustrated in Figure 1 (resp. Fig. 2). Here, we observe that in all cases, the textbook approach is always the worst and that, except for the second order, the (non-corrected) two-pass and pairwise approaches exhibit similar relative errors. Therefore, the major advantage in terms of execution speed, inherent to our one-pass incremental approach over the two-pass, is not offset by degraded numerical accuracy. It should be noted that empirical results are calculated using single-precision arithmetic, and thus an accuracy breakdown near 1.2×10^{-7} is unavoidable. We nevertheless observe that when numerical accuracy takes precedence over execution speed, and when passing twice over the data is feasible, a corrected two-pass algorithm always fares best. Our arbitrary-order correction provided in (3.40) should be used in this case.

4.3 Application to a scientific use case

Finally, we demonstrate the benefit of employing the single-pass update algorithm in the statistical analysis of a real scientific use-case; turbulent combustion. First-principles direct numerical simulations (DNS) of turbulent combustion provide fundamental insight into the nature of turbulence-chemistry interactions which are highly non-linear and span a broad range of spatio-temporal scales (Chen et al. 2009).

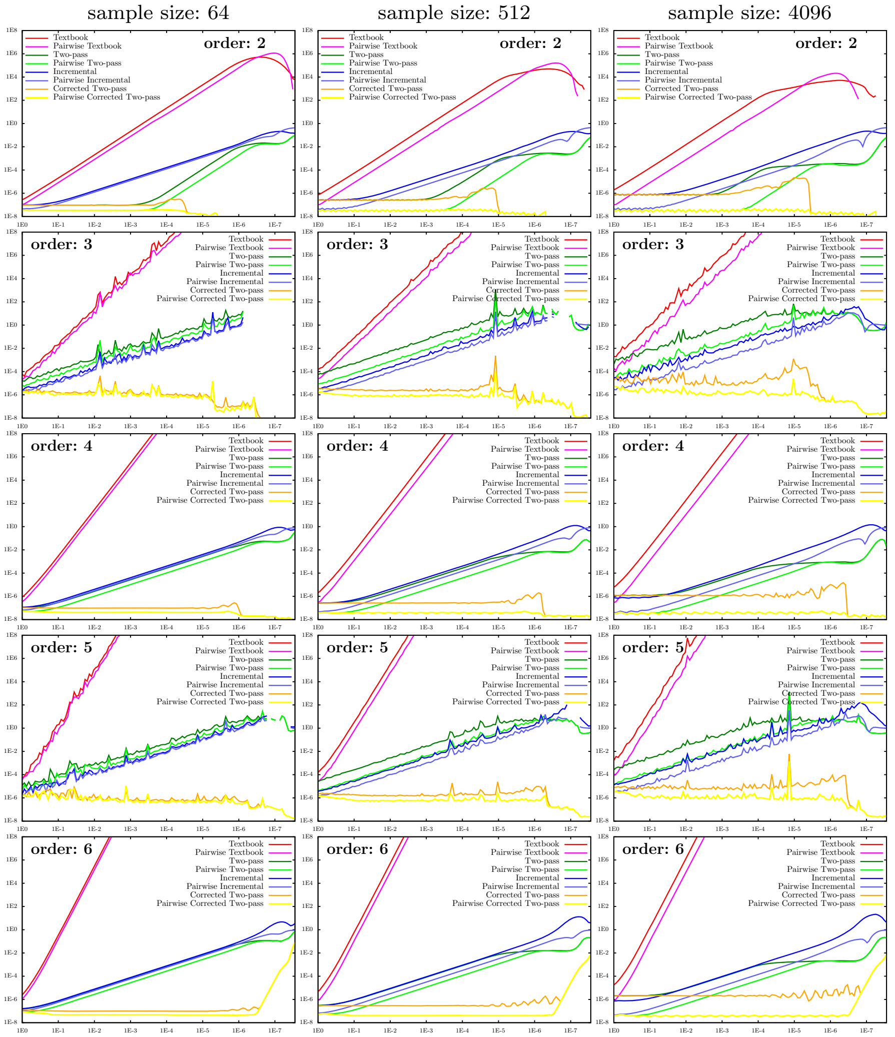

Fig. 1 The relative error (ordinate) in moments of orders 2 through 6 averaged over 2^{16} samples of size 64 (*left*), 512 (*center*), and 4096 (*right*) drawn from a Gaussian distribution with mean 1 and varying standard deviation (abscissa)

Statistical moments, both weighted and unweighted, are often used to quantify and describe the nature of turbulence-chemistry interactions from DNS.

To evaluate the performance of the proposed algorithm we compare the analysis time taken to extract second moments from a real combustion DNS dataset. The DNS corresponds to a statistically-stationary premixed rectangular slot-jet Bunsen flame of methane-air mixture (Sankaran et al. 2007). The DNS is performed on a three-dimensional rectangular Cartesian domain consisting of 720 × 400 × 180 grid points in x, y and z directions respectively and the solution vector consists of 18 variables at each grid point. It is often of interest in simulations like these to study the moments of

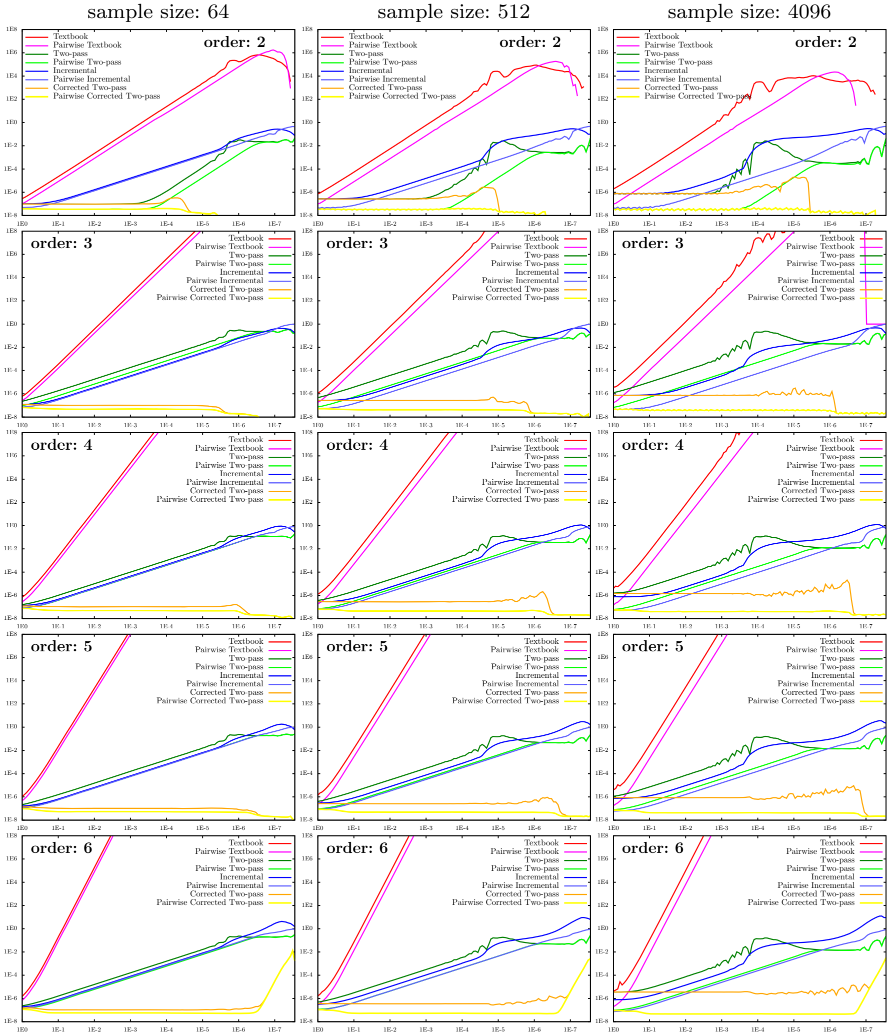

Fig. 2 The relative error (ordinate) in moments of orders 2 through 6 averaged over 2^{16} samples of size 64 (*left*), 512 (*center*), and 4096 (*right*) drawn from a noncentral exponential distribution with mean 1 and varying standard deviation (abscissa)

the normalised fuel mass fraction ϕ_f which is an indicator of reaction progress. The variable ϕ_f is defined such that its value goes smoothly from 0 in the unburnt state to 1 in the fully burnt state.

Sixty temporal snapshots of the DNS were analysed on 600 cores of a CRAY cluster. The configuration time and z direction are statistically homogeneous and hence moments are constructed by loading each snapshot of data and performing MPI collectives along the z direction. By far the biggest bottleneck in such a workflow is the time taken to load each snapshot of data from disk using parallel I/O since the MPI collectives finish much quicker in comparison. For this DNS it took approximately

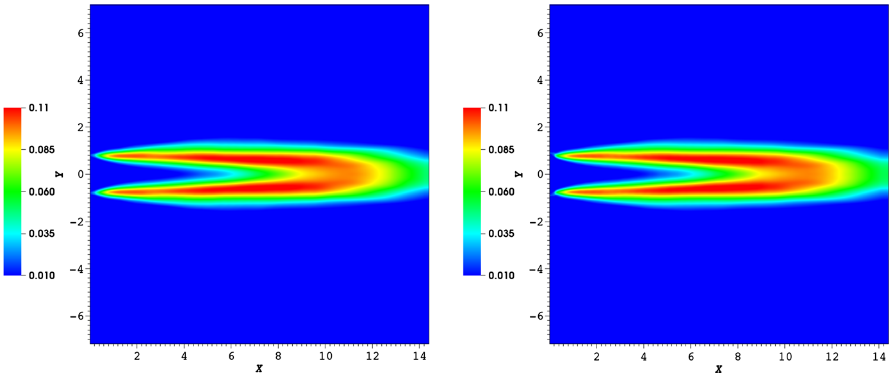

Fig. 3 Comparison of the compound second moment of ϕ_f extracted from turbulent combustion DNS data. *Left* moment extracted using the single-pass algorithm. *Right* moment extracted using the naive two-pass algorithm

11 s to load one snapshot while the compute time, including the collectives, for each snapshot was a few milliseconds. This trend will worsen at extreme scale where nodes are projected to have large concurrency and data movement is projected to be very expensive, both in terms of time and power consumption.

For the current analysis, unsurprisingly, extracting second moments using the single-pass algorithm took roughly half the amount of time (≈660 s) compared to the naive two-pass algorithm (≈1320 s) on the same amount of computational resources (600 cores). Figure 3 shows the compound moment of ϕ_f extracted using both the single-pass and two-pass algorithms. The results are numerically very close and the Euclidean norm of the difference is 4.4×10^{-4} (note that ϕ_f is a quantity which is of order unity). Again, we observe that the gain in efficiency of our incremental one-pass approach is not at the expense of numerical accuracy.

5 Conclusion

In this work we started by surveying existing results for the derivation of incremental and parallel formulas for univariate and multivariate moments of arbitrary order, and discussed their limitations in terms of numerical stability and accuracy as well as for the specific aim of massively parallel calculations. Subsequently, we derived generalized, arbitrary-order, arbitrary-variate one pass formulas which exhibit numerical stability while lending themselves to scalable parallel implementations. Furthermore, we developed extensions of these results for the computation of weighted or compound moments, and demonstrated their use in the numerical analysis of turbulent flows. In addition, we have developed a generalized correction factor for the standard two-pass moment algorithms, that supports the maintenance of accuracy over nearly the full representable range of the input, avoiding the need for extended precision arithmetic.

We also provided the results from a suite of empirical tests to study the condition number and relative error of various formulations. Our findings show that the generalized incremental and pairwise algorithms perform almost as well as the standard two-pass formulations, and in some cases even better. Thus, the reduction in execution speed inherent to the one-pass incremental approach over the two-pass is generally not offset by degraded numerical accuracy. We observed in our experiments that a corrected two-pass algorithm using our arbitrary-order correction provided in (3.40), results in the smallest overall error. This approach is only necessary when numerical accuracy takes precedence over execution speed, and when passing twice over the data is feasible.

References

Amblard PO, Brossier JM (1995) Adaptive estimation of the fourth-order cumulant of a white stochastic process. Signal Process 42(1):37–43

Bennett J, Grout RW, Pébay PP, Roe DC, Thompson DC (2009) Numerically stable, single-pass, parallel statistics algorithms. In: CLUSTER. IEEE, pp 1–8. doi:10.1109/CLUSTR.2009.5289161

Chan TF, Lewis JG (1978) Rounding error analysis of algorithms for computing means and standard deviations. Tech. Rep. Tech. Rep. 289, Dept. of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD

Chan TF, Golub GH, LeVeque RJ (1979) Updating formulae and a pairwise algorithm for computing sample variances. Technical Report STAN-CS-79-773, Stanford University, Department of Computer Science

Chan TF, Golub GH, LeVeque RJ (1983) Algorithms for computing the sample variance: analysis and recommendations. Am Stat 37(3):242–247

Chen JH, Choudhary A, de Supinski B, DeVries M, Hawkes ER, Klasky S, Liao WK, Ma KL, Mellor-Crummey J, Podhorski N, Sankaran R, Shende S, Yoo CS (2009) Terascale direct numerical simulations of turbulent combustion using S3D. Comput Sci Discov 2(015001):1–31

Dembélé D, Favier G (1998) Recursive estimation of fourth-order cumulants with application to identification. Signal Process 68(2):127–139

Dodge Y, Rousson V (1999) The complications of the fourth central moment. Am Stat 53(3):267–269

Eddelbuettel D (2010) High-performance and parallel computing with R. http://cran.r-project.org/web/views/HighPerformanceComputing.html, version 2010-09-16 downloaded 2010-11-15

Farid H, Popescu AC (2001) Blind removal of image non-linearities. In: Proceedings of the 8th international conference on computer vision (ICCV'01), vol 1, Vancouver, BC, pp 76–81

Halmos PR (1946) The theory of unbiased estimation. Ann Math Stat 17(1):34–43

Harvey CR, Siddique A (2000) Conditional skewness in asset pricing tests. J Financ 55(3):1263–1295

Hung DCH, Shackleton M, Xu X (2004) CAPM, higher co-moment and factor models of UK stock returns. J Bus Financ Acc 31(1–2):87–112

Ibrahim MA, Hussein AWF, Mashali SA, Mohamed AH (1998) A blind image restoration system using higher-order statistics and Radon transform. In: Proceedings of the 5th international conference on electronics, circuits and systems (ICECS'98), vol 3, Lisboa, Portugal, pp 523–530

IEEE (1985) IEEE standard for binary floating-point arithmetic. ANSI/IEEE Std 754-1985

Jones WP (1993) Turbulence modelling and numerical solution methods for variable density and combusting flows. In: Libby PA, Williams FA (eds) Turbulent reacting flows. Academic Press, London, pp 309–374

Kikuchi N, Hayase S, Sekine K, Sasaki S (2005) Performance of chromatic dispersion monitoring using statistical moments of asynchronously sampled waveform histograms. Photonics Technol Lett 17:1103–1105

Kleihorst RP, Lagendijk RL, Biemond J (1997) An adaptive order-statistic noise filter for gamma-corrected image sequences. IEEE Trans Image Process 6(10):1442–1446

Langlois P, Louvet N (2007) How to ensure a faithful polynomial evaluation with the compensated horner algorithm. In: Proceedings of the 18th symposium on computer arithmetic (ARITH'07), Montpellier, France, pp 141–149

Lyu S, Farid H (2002) Detecting hidden messages using higher-order statistics and support vector machines. In: Proceedings of the 5th international workshop on information hiding (IH'02), vol 2578, Lecture Notes in Computer Science. Springer, Noordwijkerhout, The Netherlands, pp 340–354

Mendel JM (1991) Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications. Proc IEEE 79(3):278–305

Neely PM (1966) Comparison of several algorithms for computation of means, standard deviations and correlation coefficients. Commun ACM 9(7):496–499

Nemer E, Goubran R, Mahmoud S (2001) Robust voice activity detection using higher-order statistics in the LPC residual domain. IEEE Trans Speech Audio Process 9(3):217–231

Nikias CL, Mendel JM (1993) Signal processing with higher-order spectra. IEEE Signal Process Mag 10(3):10–37

Ogita T, Rump SM, Oishi S (2005) Accurate sum and dot product. SIAM J Sci Comput 26(6):1955–1988

Pearson K (1894) Contributions to the mathematical theory of evolution. Philos Trans R Soc Lond Ser A 185:71–110

Pébay P, Thompson DC, Bennett J (2010) Computing contingency statistics inparallel: design trade-offs and limiting cases. In: Proceedings of the 2010 IEEE international conference on cluster computing (12th CLUSTER'10). IEEE Computer Society, Heraklion, Crete, Greece, pp 156–165

Pébay P, Thompson DC, Bennett J, Mascarenhas A (2011) Design and performance of a scalable, parallel statistics toolkit. In: Proceedings of the workshop on parallel & distributed scientific & engineering computing (12th PDSEC'11) workshop proceedings, 25th IEEE international symposium on parallel and distributed processing (25th IPDPS'11). IEEE Computer Society, Anchorage, Alaska, USA, pp 1475–1484

Porat B, Friedlander B (1991) Direction finding algorithms based on high-order statistics. IEEE Trans Signal Process 39(9):2016–2024

Prakasam P, Madheswaran M (2008) M-ary shift keying modulation scheme identification algorithm using wavelet transform and higher order statistical moment. J Appl Sci 8(1):112–119

Samuelson PA (1970) The fundamental approximation theorem of portfolio analysis in terms of means, variances and higher moments. Rev Econ Stud 37(4):537–542

Sankaran R, Hawkes ER, Chen JH, Lu T, Law CK (2007) Structure of a spatially developing turbulent lean methaneair Bunsen flame. Proc Combust Inst 31:1291–1298

Schmidberger M, Morgan M, Eddelbuettel D, Yu H, Tierney L, Mansmann U (2009) State-of-the-art in parallel computing with R. Tech. Rep. 47, Department of Statistics, University of Munich

Shalvi O, Weinstein E (1990) New criteria for blind deconvolution of nonminimum phase systems (channels). IEEE Trans Inf Theory 36(2):312–321

Stata (2010) Stata/MP performance report. Tech. rep., StataCorp LP. http://www.stata.com/statamp/statamp.pdf, version 2.0.0

Tugnait JK (1997) Identification and deconvolution of multichannel linear non-Gaussian processes using higher order statistics and inverse filter criteria. IEEE Trans Signal Process 45(3):658–672

Wang H, Huo J, Song C (2006) A blind image restoration algorithm based on cumulants. In: Proceedings of the 8th international conference on signal processing (ICSP'06), Beijing

Welford BP (1962) Note on a method for calculating corrected sums of squares and products. Technometrics 4(3):419–420

West DHD (1979) Updating mean and variance estimates: an improved method. Commun ACM 22(9):532–535

Wong MH, Thompson DC, Pébay P, Mayo JR, Gentile AC, Debusschere BJ, Brandt JM (2008) OVIS-2: a robust distributed architecture for scalable RAS. In: Proceedings of the 22nd IEEE international parallel & distributed processing symposium, Miami, FL. http://ipdps.org/ipdps2008/IPDPS-2008-Abstract.pdf

Wylie B, Baumes J, Shead T (2008) Titan informatics toolkit. In: IEEE visualization tutorial, Columbus, OH

Xu Y, Crebbin G (1996) Image blur identification by using higher order statistic techniques. In: Proceedings of the 3rd international conference on image processing (ICIP'96), vol 3, Lausanne, Switzerland, pp 77–80