Testing the numerical correctness of software for the calculation of sample standard deviation

M G Cox, M P Dainton and P M Harris

National Physical Laboratory, Teddington, Middlesex TW11 0LW, UK

1 Abstract

We describe the application of a general methodology for testing the numerical accuracy of software to functions for the calculation of the sample standard deviation. Each stage of the methodology, from the provision of a specification for the function tested through the definition of performance parameters and measures to the presentation and interpretation of the test results, is presented. In this way, and by stating any assumptions made in the application of the methodology, the testing undertaken is made as objective as possible.

2 Introduction

The numerical correctness of scientific software is important to metrology because a poor software implementation can lead to inaccuracy that could have been avoided, and as a consequence the correctness of measurement results can be compromised. Although there is some awareness of the potential limitations of spreadsheets and other software packages arising from the use of inaccurate or unstable numerical algorithms, relatively little testing and validation is performed on such software. Often, there is a tendency to rely upon well-established packages and to perform only a small number of checks using alternative methods of calculation. As part of the first National Measurement System Software Support for Metrology programme, NPL has undertaken work to develop an infrastructure, comprising supporting information and guidelines, to ensure that the use of software within metrology, particularly spreadsheets and proprietary software packages, is made as effective as possible. For example, one output of this work has been a number of reports that describe the objective testing of the intrinsic and in-built functions included within spreadsheets and other proprietary software packages that are routinely used in metrology applications.

Here, we describe the application of a general methodology for testing the numerical correctness of software to the calculation of the sample standard deviation taken from a number of spreadsheet, statistical and scientific software packages, including Microsoft Excel, MathCAD, S-PLUS, Matlab, and the NAG and IMSL Fortran libraries. The basis of the approach is the design and use of reference data sets and corresponding results to undertake black-box testing of the software under investigation. The reference data sets and results are generated in a manner consistent with the functional specification of the test software, and the results returned by the test software for the reference data sets are then compared objectively with the reference results using performance measures. Finally, the performance measures are interpreted in order to decide whether the test software meets requirements and is fit for purpose.

3 Sample standard deviation

Consider the calculation of the *sample standard deviation* s defined, for the sample $\{x_1, x_2, \ldots, x_m\}$, by

$$s = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(x_i - \bar{x}\right)^2},$$

where $\bar{x}$ is the sample mean,

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i .$$

The calculation of the sample standard deviation is important to metrology because of its role as a statistic for estimating the standard deviation of the population from which the sample was drawn, and within uncertainty evaluation as the basis for understanding the repeatability of a measurement process.

4 Methodology

A general methodology for evaluating the numerical accuracy of the results produced by scientific software has been developed [1]. For a *user* of a proprietary software package, the methodology involves the following six stages:

1. documenting the specification of the test software,

2. interfacing to the test software,

3. specification of reference data sets,

4. specification of performance measures and testing requirements,

5. generation of reference data sets and corresponding reference results, and

6. presentation and interpretation of performance measures.

Each stage in the general approach, and the manner in which it is applied to testing functions for the calculation of the sample standard deviation, is given in the following sections.

5 Specification of the test software

For testing of software to be meaningful, it is necessary to have a full and unambiguous specification of the task carried out by the software. If the specification is inconsistent with this task, testing in accordance with the specification might yield the conclusion that the software was deficient, when in fact it might be acceptable.

The particular implementations of the sample standard deviation function tested in this work are those contained within the following packages: the IMSL Fortran library [3], the NAG Fortran library [4], Matlab [5], Microsoft Excel [6], S-PLUS [7], and MathCAD [8]. Detailed software specifications of the functions tested, including descriptions of their input and output parameters, are available [2].

6 Interfacing to the test software

Where possible the testing procedure is executed automatically using for each package its own intrinsic programming language: VBA (Visual Basic for Applications) for Excel, Fortran for the NAG and IMSL libraries, MathCAD's own programming syntax, etc. The test functions for sample standard deviation are called in such a way as to mimic how a user might be expected to access the functions. Each reference data set, with its corresponding reference result and other information, is read from data files. Values of the performance parameter are calculated using functions provided by the intrinsic programming language, written to data files, and displayed using Excel's graphing facilities.

7 Specification of reference data sets

Performance parameters are used to capture the properties of data sets that would be encountered in practice and to describe the range of admissible inputs to the test software. Sequences of data sets may be generated by varying an individual performance parameter. A sequence forms a *graded* sequence in a case where the performance parameter relates directly to the condition or "degree of difficulty" of the problem represented by the data. The degree of difficulty measures the change in the result relative to a small change in the data (for a perfect implementation). By investigating the performance of the test software for such graded sequences, it is possible to identify cases where the test software is based upon a poor choice of mathematical algorithm.

Performance parameters for the sample standard deviation include the reference sample mean $\bar{x}$, the reference sample standard deviation s^{ref}, and the number m of points in the sample.

8 Specification of performance measures

Performance measures are used to quantify the performance of the test software for the reference data sets to which the test software is applied. Furthermore, by relating the values of these metrics to the requirements of the user, it is possible to assess objectively whether the test software meets these requirements and is therefore fit for purpose.

A performance measure is used that accounts for (a) the error in the computed test result relative to the reference result, (b) the "degree of difficulty" of the problem represented by the data, and (c) the precision of the arithmetic used to compute the test result. The performance measure $P(s)$ used to measure the departure of the computed sample standard deviation s returned by the test software from the reference sample standard deviation s^{ref} is given by

$$P(s) = \log_{10}\left(1 + \frac{1}{\kappa(s)\eta}\frac{\left|s - s^{\mathrm{ref}}\right|}{\left|s^{\mathrm{ref}}\right|}\right),$$

where $\kappa(s)$ measures the problem degree of difficulty,

$$\kappa(s) = \max\left\{\frac{|\bar{x}|}{s^{\mathrm{ref}}}, 1\right\},$$

and η is the machine precision [1]. The performance measure $P(s)$ indicates the number of figures of accuracy lost by the test software over and above what software based on an optimally stable algorithm would produce using machine precision η.

9 Generation of reference data sets and corresponding reference results

In the general methodology, a *reference pair*, i.e., a reference data set and corresponding reference results, may be produced either using *reference software* or a *data generator* [1]. Reference software is software written to an extremely high standard to solve the problem given in the specification of the test software. Alternatively, data generators are used to construct reference data sets with known solutions, i.e., solutions specified *a priori*, and are based on methods, such as *null-space methods* [1], that use a solution characterisation to construct families or classes of data sets possessing nominally the same solution. For the testing of functions for calculating the sample standard deviation a data generator, implementing the null-space method, was used [2].

10 Presentation and interpretation of results

Figure 1 illustrates the performance of each sample standard deviation function tested for a graded sequence of reference data sets obtained by setting the reference sample standard deviation to unity, the sample size to 100, and allowing the sample mean (the performance parameter) to take the values 10^k, $k = 0, \ldots, 16$. It shows, for each test function, the performance measure $P(s)$ plotted against the sample mean of each reference data set within this sequence.

The graph indicates that the IMSL, NAG, Matlab, S-PLUS and MathCAD functions give results that are accurate for the sequence of reference data sets considered, never losing more than two digits of accuracy than would a reference implementation. However, the results for the Excel function show that this function loses up to eight additional figures of accuracy. A clear trend in the performance of the test software as a function of the reference sample mean (and hence the degree of difficulty) is apparent for this function.

A detailed analysis of the results obtained from testing functions for calculating sample standard deviation using a number of sequences of reference data sets is available [2]. The results indicate that the IMSL, NAG, Matlab, S-PLUS and MathCAD packages provide reliable software for the calculation of the sample standard deviation. However, this is not the case for the Excel spreadsheet package, which appears to implement an algorithm whose performance degrades as a function of the problem degree of difficulty. For general use of Excel's standard deviation function, users are recommended to pre-process the data (using mean-centring of the data) prior to application of the Excel function.

11 Conclusions

A general methodology for testing the numerical accuracy of scientific software has been applied to functions for the calculation of sample standard deviation taken from a range of software packages. The calculation of sample standard deviation was chosen as it is one of the most commonly-used functions within scientific applications and especially metrology. At first sight its implementation might be thought to be straightforward. However this has been shown not to be the case, and the need for rigorous testing has been demonstrated.

The rigorous testing of software used in metrology is particularly important because metrology computations frequently require statistics of *specially structured data*, e.g., the sample mean and standard deviation of a set of nearly identical numbers (such as arise in a series of replicate weighings). For such circumstances, unstable algorithms are likely to deliver unreliable results, and thus considerations of this type need to be taken into account in developing special methods of test for spreadsheets used in metrology.

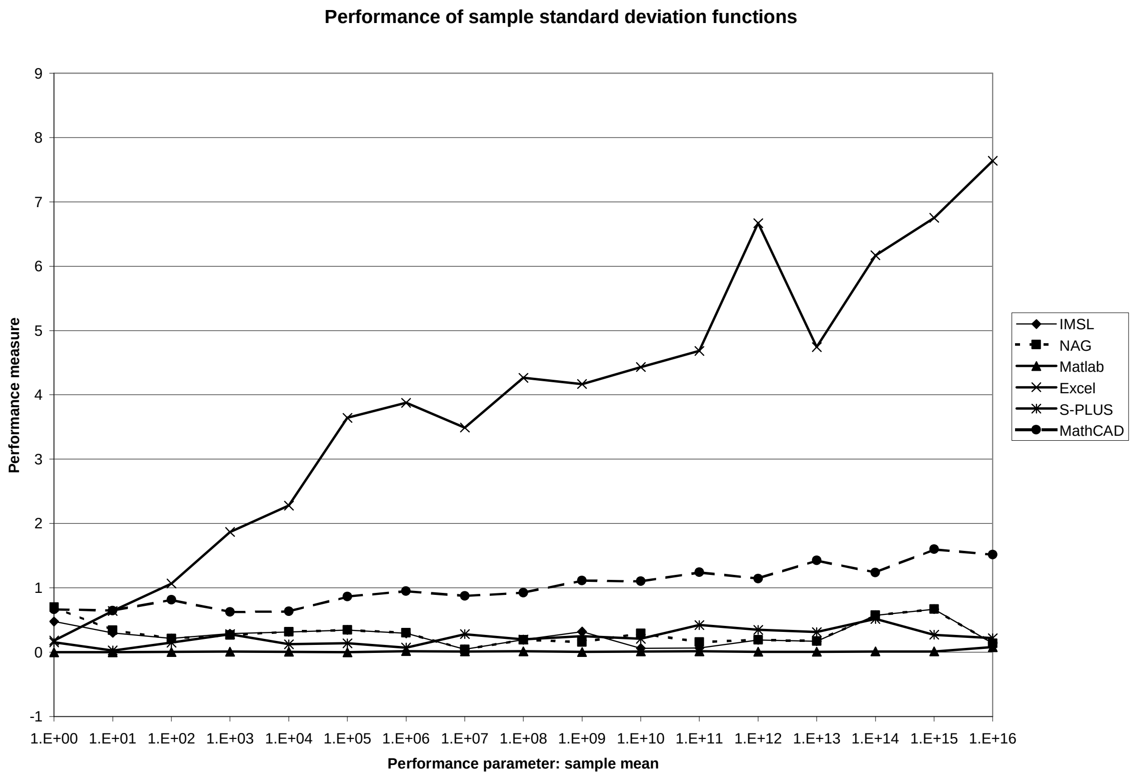

Figure 1 Performance of each sample standard deviation function for a graded sequence of reference data sets for which the performance parameter is the sample mean. The performance measure is the number of significant figures lost by the function compared with a reference implementation.

12 Acknowledgements

This report constitutes one of the deliverables of the second National Measurement System Software Support for Metrology programme (April 2001 to March 2004), and has been funded by the National Measurement System Policy Unit of the UK Department of Trade and Industry.

The authors would like to thank Jessica Barrett and Paul Kenward for their help in carrying out the testing described here.

13 References

1. H.R. Cook, M.G. Cox, M.P. Dainton and P.M. Harris. A methodology for testing spreadsheets and other packages used in metrology. NPL Report CISE 25/99, September 1999.

2. M.G. Cox, M.P. Dainton and P.M. Harris. Testing spreadsheets and other packages used in metrology: Testing functions for the calculation of standard deviation. the intrinsic functions of MathCAD. NPL Report CMSC 07/00, October 2000.

3. IMSL Fortran 90 Math/Library (version 3.0) provided with Digital Visual Fortran (version 5.0.D), Professional Edition, Digital Equipment Corporation.

4. The NAG Fortran Library Manual (Mark 19), Numerical Algorithms Group.

5. Matlab Version 5.3.0.10183 (R11). Copyright ©1984–1999, The MathsWork Inc.

6. Microsoft Excel 2000. Copyright © 1985–1999 Microsoft Corporation.

7. S-Plus 4.0 Release 3 for Windows Copyright ©1988–1997, MathSoft Inc.

8. MathCad Version 8, Professional. Copyright © 1986–1998, MathSoft Inc.