

ARE OCTAVE, SCILAB AND MATLAB RELIABLE?

Alejandro C. Frery^{a,b}, Eliana S. de Almeida^{a,b} and Antonio C. Medeiros^b

^a*Laboratório de Computação Científica e Análise Numérica (LaCCAN), Centro de Pesquisas em Matemática Computacional (CPMAT), Instituto de Computação, Universidade Federal de Alagoas, BR 104 Norte km 97, 57072-970 Maceió, AL – Brazil*

^b*Laboratório de Computação Científica e Visualização (LCCV), Centro de Tecnologia, Universidade Federal de Alagoas, BR 104 Norte km 97, 57072-970 Maceió, AL – Brazil*

Keywords: Numerical analysis, computational platforms, statistical computing, spectral graph analysis

Abstract. In a word, no. In this article we show evidence that three platforms frequently used for solving Engineering problems have numerical pitfalls. These platforms are Octave, Scilab and Matlab, running on i386 architecture and three operating systems (Windows, Ubuntu and Mac OS). They were submitted to two comprehensive tests, namely the data sets and functions provided by NIST (National Institute of Standards and Technology), and our proposal of a set of matrices and operations on them. NIST protocol includes the computation of basic univariate statistics (mean, standard deviation and first-lag correlation), linear regression, and extremes of probability distributions. Our set of operations include matrix inversion and the computation of the determinant and eigenvalues. The assessment is made comparing the results computed by the platforms, and assessing the number of correct digits with respect to certified values. Serious pitfalls are identified in seemingly easy tasks as, for instance, the first-lag auto-correlation coefficient. Whenever available, the results are compared with those provided by R, a FLOSS (Free/Libre Open Source Software), whose excellent numerical abilities have been reported elsewhere.

1 INTRODUCTION

Numerical computation of both continuous and discrete mathematical problems is at the core of all scientific and technological activities. At the heart of this approach to solving problems lies the search for practical approximate solutions within reasonable bounds on errors.

Numerical computing had a dramatic growth with the emergences of libraries as, for instance, IMSL and NAg. Such libraries provide high quality, tested routines for numerical analysis, allowing researchers to focus on the application rather than on the tool. The rationale for the existence of such libraries is that certain low level problems are common to many scientific disciplines as, for instance, computing basic statistics, matrix operations, zeros of polynomials and integration.

The success of those libraries led to the development of numerical platforms, with languages of their own that simplified the development of prototypes. These platforms frequently replace the direct use of C or Fortran numerical libraries.

Little attention has been drawn to systematic testing of such platforms in the diversity of hardware and operating systems they are offered. Examples of such assessments, but limited to spreadsheets, have been conducted leading to a vast literature. Most of these studies follow the same methodology: constasting results with the certified values provided by benchmarks. Among the last, the datasets provided by the [National Institute of Standards and Technology](#) are widely employed. We use some of those datasets, and propose tests that employ operations on matrices.

Three numerical platforms were tested here: Octave 3.2.3, Scilab 5.2.2 and Matlab 7.9.0.529 (R2009b). Whenever available, they were checked in three operating systems: Windows XP Professional SP 2, Linux Ubuntu 10.4 and Mac OS X Leopard 10.5.6. In all cases, i386 architecture hardware was employed, and double precision computation was enforced.

The paper unfolds as follows. Section 2 discusses briefly how accuracy is measured in the remainder of the work. Section 3 presents the results obtained assessing basic statistics (Section 3.1), probability distribution functions (Section 3.2), linear regression (Section 3.3) and operations on matrices (Section 3.4). Section 4 concludes the paper.

2 MEASURING ACCURACY

Three sources of numerical errors can be introduced in the solution of a problem, namely (i) round-off, (ii) truncation and discretization, and (iii) numerical instability.

The availability to implementation details is limited in most commercial software and, even if the algorithms were available, other factors have impact on the software accuracy (implementation details, hardware, compiler, operational system etc.). Due to such limitations, we followed the strategy adopted by many authors (c.f. [Knüsel, 2005](#); [Kruck, 2006](#); [McCullough and Heiser, 2008](#), for instance), which consists of adopting a user viewpoint: datasets with known properties are used as input, and the results provided by the software under assessment is contrasted with that known to be correct.

Two measures of accuracy will be employed. The first is the base-10 logarithm of the absolute value of the relative error:

$$\text{LRE}(x, c) = \begin{cases} -\log_{10} \frac{|x-c|}{|c|} & \text{if } c \neq 0, \\ -\log_{10} |x| & \text{otherwise,} \end{cases} \quad (1)$$

where x is the value computed by the software and c is the certified value. LRE (*Log-Relative Error*) relates to the number of significant digits that were correctly computed. We will adopt the

convention of reporting one decimal place whenever $LRE(x, c) \geq 1$, zero if $0 \leq LRE(x, c) < 1$ and “-” otherwise; there is no worse situation than “-”. An exact match is reported as ‘Inf’, and ‘NA’ denotes that the platform returned an error (segmentation fault or similar). The LRE is computed R (<http://www.r-project.org/>), whose excellent numerical properties were checked by Almiron et al. (2009).

The second measure of accuracy will be employed in Section 3.4, where it is known that certain values are equal to zero. In such cases, the result of the logical comparison of the computed value and zero in the platform under assessment is the measure of accuracy. The rationale behind this choice is that the interest in such cases is not the value itself, but when it is zero it indicates situations of interest.

In all cases double precision was employed.

3 RESULTS

3.1 Basic Statistics

Almiron et al. (2010) showed that simple descriptive statistics pose difficulties for widely used spreadsheets. The authors considered the sample mean, the sample standard deviation and the sample first lag correlation computed on nine datasets from the *Statistical Reference Datasets* from the (American) National Institute of Standards and Technology (NIST). These datasets are classified in three levels of numerical difficulty: *low*, *average* and *high*. The certified values were calculated using multiple precision arithmetic to obtain 500 digits answers.

The datasets with low difficulty are *Lew*, *Lottery*, *Mavro*, *Michelso*, *NumAcc1* and *PiDigits*. *NumAcc2* and *NumAcc3* are average difficulty datasets and *NumAcc4* is the only high difficulty dataset for univariate summary statistics.

The mean is produced in all platforms by the commands `mean`. The standard deviation in Octave and MatLab is computed with the command `std`, whereas the SciLab command is `st_deviation`. Only SciLab provides a native function for computing the correlation, namely `correl`; in the other two platforms it has to be computed. For the sake of compatibility of the results, the command `Correl(v(1:n-1), v(2:n))` in Octave and the command `corr(v(1:n-1), v(2:n))` in Matlab are used to compute the the first lag correlation of the vector v of size $n \geq 2$, respectively. Note that the use of spectral methods is precluded in this case.

Table 1 presents the accuracy of the three programming ambients under assessment running (whenever available) under Windows (‘Win’), Linux (‘Lin’) and Mac OS (‘Mac’).

3.2 Statistical Functions

Statistical functions play a central role in data analysis procedures. Many of them employ special functions and integration algorithms.

One of the most complete numerical assessments of such procedures was performed by Yalta (2008), but limited to Microsoft’s Excel[®]. In the following, we present the numerical accuracy of the routines provided by Octave, SciLab and MatLab in those cases that were identified as problematic in that work. Those results obtained with the Mathematica 5.2 software by Yalta (2008), which are certified to be accurate to six significant digits for all the distribution functions here assessed.

The distributions herein assessed are the binomial (Table 2), Poisson (Tables 3 and 4), gamma (Table 5), normal (Table 6), χ^2 (Table 7), beta (Table 8), t-Student (Table 9) and F (Table 10).

Scilab provides the commands `cdfbin`, `cdfpoi`, `cdfgam`, `cdfnor`, `cdfbet`, `cdft` and

Table 1: LREs for the sample mean, standard deviation and first lag correlation

Platform	OS	PiDigits	Lottery	Lew	Mavro	Michelso	Numacc1	Numacc2	Numacc3	Numacc4
Sample Mean										
Octave	Windows	Inf	15.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf
	Linux	Inf	5.6	4.6	4.7	5.1	Inf	Inf	6.7	7.7
	Mac Os	Inf	Inf	Inf	Inf	Inf	Inf	14.0	Inf	14.0
MatLab	Windows	Inf	15.0	Inf	Inf	15.0	Inf	14.0	15.0	14.0
Scilab	Windows	Inf	8.1	Inf	Inf	Inf	Inf	Inf	Inf	7.7
	Linux	Inf	8.1	Inf	Inf	Inf	Inf	Inf	Inf	7.7
	Mac OS	Inf	8.1	Inf	Inf	Inf	Inf	Inf	Inf	7.7
Sample Standard Deviation										
Octave	Windows	15.0	Inf	15.0	13.0	14.0	Inf	15.0	9.0	8.0
	Linux	4.9	6.0	5.1	5.1	5.1	Inf	Inf	Inf	Inf
	Mac Os	Inf	Inf	Inf	13.1	13.8	Inf	Inf	9.5	8.3
MatLab	Windows	14.9	5.0	15.0	12.9	14.0	Inf	Inf	9.5	8.3
Scilab	Windows	7.9	8.1	8.2	4.1	6.2	Inf	Inf	Inf	Inf
	Linux	7.9	8.1	8.2	4.1	6.2	Inf	Inf	Inf	Inf
	Mac OS	7.9	8.1	8.2	4.3	6.2	Inf	Inf	Inf	Inf
Sample First Lag Correlation										
Octave	Windows	4.0	2.0	3.0	2.0	4.0	2.0	3.0	3.0	3.0
	Linux	3.9	2.1	2.6	1.8	3.6	0	3.3	3.3	3.3
	Mac Os	3.0	2.0	2.6	1.7	3.6	0	3.0	3.0	3.0
MatLab	Windows	4.0	2.1	2.6	1.8	3.6	0	3.3	3.3	3.3
Scilab	Windows	0	-	-	0	0	0.5	0	0	0
	Linux	0	-	-	0	0	0.5	0	0	0
	Mac OS	0	NA	NA	0	0	0.5	0	0	0

`cdf` to compute the binomial, Poisson, gamma, normal, beta, t-Student and F , respectively. Matlab and Octave use the same commands to compute all the distributions except for beta. The commands are `binocdf` for binomial, `poisscdf` for Poisson, `gamcdf` for gamma, `norminv` for normal, `tinv` for t-Student and `finv` for F . The beta distribution is computed in Octave with the command `beta_cdf` and Matlab provides the command `betainv`.

The computed quantities are probabilities, cumulative distribution functions, and quantiles.

3.3 Linear Regression

The eleven datasets employed in this section span cases of low (NORRIS and PONTIUS), average (NOINT1 and NOINT2) and high numerical difficulty (the other seven datasets). Each dataset is used to perform a linear regression with a specified number of coefficients. Table 11 presents the smallest LRE of each regression coefficient, and the LRE of the residual standard deviation (RSD) of each fit.

Octave does not provide an explicit function for performing linear regression. Rather than that, linear regression is computed solving a least squares problem, and the data requires prior preparation for that. Scilab provides the function `reglin` to obtain the β coefficients and RSD.

3.4 Decisions based on Matrices

The first assessment of matrix computing we propose is the simplest one: computing the determinant of a 2×2 matrix. Consider the matrix

$$M = \begin{pmatrix} b & b\varepsilon \\ s/\varepsilon & s \end{pmatrix},$$

which clearly has null determinant $|M| = 0$. We force the numerical computation of the determinant $|\widetilde{M}|$ with built-in functions, ensuring that the intermediate values $(b\varepsilon)$ and (s/ε) are evaluated. In order to check the accuracy of the platforms, the values herein assessed are $b = 10^j$ and $s = 10^{-j}$, with $j \in \{0, 1, \dots, 15\}$, and $\varepsilon = 0.\underbrace{9\dots9}_{k \text{ times}}$, $k \in \{1, \dots, 15\}$.

Determinants were computed using the `det` command, which is common to all platforms. Our assessment is based on the decision the user is led by the result, rather than on the result itself. This is due to the fact that more often than not what users are interested upon is a decision, and not a numerical value.

The number of correct results of comparing $|\widetilde{M}|$ with zero were 150 for Matlab, 6 for Octave in Windows and Linux, 138 for Octave in Mac OS, 146 for SciLab in Windows and in Mac OS, and 5 for SciLab in Linux.

Among the many other possible assessments of matrix computations, we chose to work with spectral graph theory. This line of research within the field of graph theory deals with the properties of the Laplacian matrix, a graph representation different from the usual adjacency matrix. The Laplacian matrix is directly related to deep properties of the graph as, for instance, connectivity (Bollobas, 1998).

The advantage using spectral graph analysis is twofold. Firstly, the matrices used for the assessment are formed by values which are not prone to numerical problems, c.f., equations (2). Secondly, the rule for defining these matrices is simple to implement. The second eigenvalue of the Laplacian of a graph with n vertices is a relevant measure which is frequently computed for large values of n .

Consider the non-directed finite graph without loops $G = (V, E)$ defined by the set of vertices $V = \{w_1, w_2, \dots, w_n\}$ and edges E . The degree of vertex w_i , denoted $\deg(w_i)$, is the

Table 2: Accuracy computing binomial cumulative distribution functions, $n = 1030$ and $p = 1/2$

k	$\Pr(X \leq k)$		Matlab			Octave			SciLab			
		Certified	Windows	Win	Lin	Mac	Win	Lin	Mac	Win	Lin	Mac
1		8.96114E-308	3.0	0	0	0	Inf	Inf	Inf			
2		4.61499E-305	2.7	0	0	0	6.0	6.0	6.0			
100		1.39413E-169	1.0	0	0	0	5.5	5.5	5.5			
300		2.91621E-42	0.4	0	0	0	6.4	6.4	6.4			
400		3.89735E-13	0.2	0	3.9	3.9	6.2	6.2	6.2			
410		3.19438E-11	0.2	6.2	6.2	6.2	5.8	5.8	5.8			

Table 3: Accuracy computing Poisson probabilities, $\lambda = 200$

k	$\Pr(X = k)$		Matlab			Octave			SciLab			
		Certified	Windows	Win	Lin	Mac	Win	Lin	Mac	Win	Lin	Mac
0		1.38390E-87	5.6	5.6	5.6	5.6	5.6	5.6	5.6			
103		1.41720E-14	1.4	1.4	1.4	1.4	1.4	0	0			
315		1.41948E-14	0	0	0	0	2.9	0	0			
400		5.58069E-36	6.4	6.4	6.4	6.4	0	0	0			
900		1.73230E-286	6.0	6.0	6.0	6.0	0	0	0			

Table 4: Accuracy computing Poisson cumulative distribution functions, $\lambda = 200$

k	$\Pr(X \leq k)$		Matlab			Octave			SciLab			
	λ	Certified	Windows	Win	Lin	Mac	Win	Lin	Mac	Win	Lin	Mac
1E+05	1E+05	0.500841	1.3	0	0	–	7.1	7.1	7.1			
1E+07	1E+07	0.500084	7.1	0	0	–	6.7	6.7	6.7			
1E+09	1E+09	0.500008	6.7	0	0	–	6.1	6.1	6.1			

Table 5: Accuracy computing gamma cumulative distribution functions, $\beta = 1$

x	α	$\Pr(X \leq x)$		Matlab			Octave			SciLab		
			Certified	Windows	Win	Lin	Mac	Win	Lin	Mac	Win	Lin
0.1	0.1		0.827552	6.5	6.5	6.5	6.5	6.5	6.5			
0.2	0.1		0.879420	6.4	6.4	6.4	6.4	6.4	6.4			
0.2	0.2		0.764435	6.3	6.3	6.3	6.3	6.3	6.3			
0.4	0.3		0.776381	6.3	6.3	6.3	6.3	6.3	6.3			
0.5	0.4		0.748019	6.2	6.2	6.2	6.2	6.2	6.2			

Table 6: Accuracy computing normal quantiles, $\mu = 0$ and $\sigma = 1$

p	Certified z_p	Matlab	Octave			SciLab		
		Windows	Win	Lin	Mac	Win	Lin	Mac
5E-1	0	Inf	Inf	Inf	Inf	16.2	16.2	16.2
1E-198	-30.0529	6.3	Inf	Inf	Inf	6.3	6.3	6.3
1E-300	-37.0471	7.0	Inf	Inf	Inf	7.0	7.0	7.0

Table 7: Accuracy computing the χ^2 distribution

p	n	$\Pr(X > x) = p$	Matlab	Octave			SciLab		
		Certified x	Windows	Win	Lin	Mac	Win	Lin	Mac
2E-1	1	1.64237	5.6	5.6	5.6	5.6	5.6	5.6	5.6
1E-7	1	28.3740	6.4	6.4	6.4	6.4	6.4	6.4	6.4
1E-7	5	40.8630	6.3	6.3	6.3	6.3	6.3	6.3	6.3
1E-12	1	50.8441	6.7	7.1	7.1	7.1	6.3	6.3	6.3
0.48	778	779.312	6.2	4.3	6.2	6.2	6.2	6.2	6.2
0.52	782	779.353	6.3	6.3	6.3	6.3	6.3	6.3	6.3

Table 8: Accuracy computing beta quantiles, $\alpha = 5$ and $\beta = 2$

p	Certified	Matlab	Octave			SciLab		
		Windows	Win	Lin	Mac	Win	Lin	Mac
1E-2	2.94314E-01	6.0	6.0	6.0	6.0	6.0	6.0	6.0
1E-3	1.81386E-01	6.1	6.1	6.1	6.1	6.1	6.1	6.1
1E-4	1.12969E-01	5.4	5.4	5.4	5.4	5.4	5.4	5.4
1E-5	7.07371E-02	6.2	6.2	6.2	6.2	6.2	6.2	6.2
1E-6	4.44270E-02	6.0	6.0	6.0	6.0	6.0	6.0	6.0
1E-7	2.79523E-02	5.9	5.9	5.9	5.9	5.9	5.9	5.9
1E-8	1.76057E-02	6.3	6.3	6.3	6.3	6.3	6.3	6.3
1E-9	1.10963E-02	5.5	5.5	5.5	5.5	5.5	5.5	5.5
1E-10	6.99645E-03	6.7	6.7	6.7	6.7	6.7	6.7	6.7
1E-11	4.41255E-03	6.7	6.7	6.7	6.7	6.7	6.7	6.7
1E-12	2.78337E-03	5.9	5.9	5.9	5.9	5.9	5.9	5.9
1E-13	1.75589E-03	6.1	6.1	6.1	6.1	6.1	6.1	6.1
1E-100	6.98827E-21	6.8	0	0	0	6.8	6.8	6.8

number of vertices that have w_i as an extreme. The degree matrix is the diagonal matrix with entries $\deg(w_i)$; denote it D . The adjacency matrix A is the $n \times n$ matrix with elements a_{ij} which take value 1 if there is an edge between w_i and w_j . The Laplacian of G , denoted $\mathcal{L}(G)$, is the difference between D and A , i.e., $\mathcal{L}(G) = D - A$.

As noted by [Fiedler \(1973\)](#), among the many remarkable properties of $\mathcal{L}(G)$, one should mention the following:

- Denote $\lambda_1, \dots, \lambda_n$ the eigenvalues of $\mathcal{L}(G)$, then $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.
- The number of zero eigenvalues is the number of connected components in the graph, then
- the second smallest eigenvalue λ_2 is called *algebraic connectivity*, and G is connected if and only if $\lambda_2 > 0$.

The traditional notion of connectedness is binary, i.e., a graph is either connected or not. Algebraic connectivity depends on the number of vertices and on the way they are connected.

This approach is a standard tool in the analysis of complex networks, and connectivity is probably the first property which is verified. It is therefore relevant to check the numerical adequacy of routines that compute eigenvalues and, in particular, how dependable is the second smallest eigenvalue of the Laplacian of a graph.

[Fiedler \(1973\)](#) derived the algebraic connectivity of some important connected graphs.

Consider the class of complete bipartite graphs. In such graphs there are two subsets of vertices, say V_1 and V_2 . The connectivity is such that there are no connections between vertices belonging to the same subset, and each vertex in subset V_i is connected to every vertex in V_j , $i \neq j$. Denote such graphs $K_{m,n}$, where m, n are the cardinality of V_1, V_2 , respectively; their Laplacian has the following form:

$$\mathcal{L}(K_{m,n}) = \begin{pmatrix} n & 0 & \cdots & 0 & -1 & \cdots & -1 \\ 0 & n & \cdots & 0 & -1 & \cdots & -1 \\ \vdots & & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & n & -1 & \cdots & -1 \\ -1 & \cdots & -1 & -1 & m & \cdots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \vdots \\ -1 & \cdots & -1 & -1 & 0 & \cdots & m \end{pmatrix}. \quad (2)$$

Figure 1 presents the $K_{5,3}$ complete bipartite graph. [Bollobas \(1998\)](#) shows that the eigenvalues of the Laplacian of a $K_{m,n}$ graph are $\lambda_1 = 0$ (with multiplicity $n - 1$), n (with multiplicity $m - 1$) and $\lambda_{m+n} = m + n$. The delicate issue of computing the pseudo-inverse of this Laplacian has been studied by [Ho and Van Dooren \(2005\)](#).

We use these results about the eigenvalues of $\mathcal{L}(K_{m,n})$ for testing the accuracy of matrix operations. Consider two special cases of complete bipartite graphs, namely the ones with almost perfect balance $K_{m,m+1}$ and those with almost worst balance $K_{2,2m-1}$, testing $m \in \{9, 99, 999\}$. These values span three sizes of graphs: small, medium and big. The assessment is based upon the observation of seven quantities: (i) the LRE of the smallest eigenvalue ($\lambda_1 = 0$) denoted ℓ_1 , (ii) the LRE of the biggest eigenvalue ($\lambda_{m+n} = m + n$) denoted ℓ_{m+n} , (iii) the LRE of the sum of the eigenvalues ($\sum_i^{m+n} \lambda_i = 2mn$) denoted ℓ_S , (iv) the minimum LRE of those eigenvalues that should take value n (there are $m - 1$ of them) denoted ℓ_n , (v) the minimum

Table 9: Accuracy computing the t-Student distribution, $n = 1$

$\Pr(X > x) = p$		Matlab	Octave			SciLab		
p	Certified x	Windows	Win	Lin	Mac	Win	Lin	Mac
1E-8	3.18310E+07	–	0	0	0	6.4	6.4	6.4
1E-11	3.18310E+10	–	0	0	0	6.4	6.4	6.4
1E-12	3.18310E+11	–	0	0	0	6.4	6.4	6.4
1E-13	3.18310E+12	–	0	0	0	6.4	6.4	6.4
1E-100	3.18310E+99	–	0	–	–	6.4	6.4	6.4

Table 10: Accuracy computing the F distribution, $n_1 = n_2 = 1$

$\Pr(X > x) = p$		Matlab	Octave			SciLab		
p	Certified x	Windows	Win	Lin	Mac	Win	Lin	Mac
1E-5	4.05285E+09	6.2	0	0	0	6.2	6.2	6.2
1E-6	4.05285E+11	6.2	0	0	0	6.2	6.2	6.2
1E-12	4.05285E+23	4.4	0	0	0	6.2	6.2	6.2
1E-13	4.05285E+25	3.2	0	0	0	6.2	6.2	6.2
1E-100	4.05285E+199	–	0	–	–	6.2	0	6.2

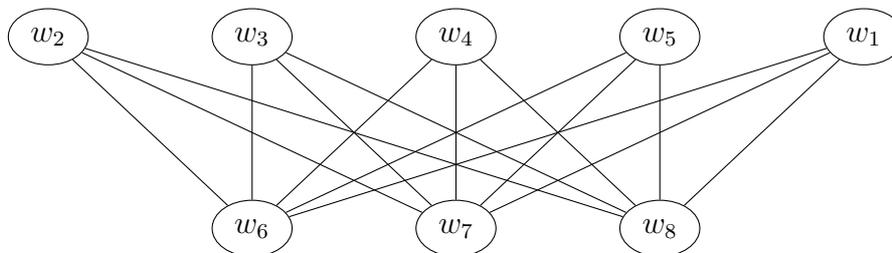


Figure 1: Complete bipartite graph $K_{5,3}$

Table 11: LRE of linear regression results

Data	Matlab			Octave			SciLab						
	Windows		Windows	Linux		Linux	Mac OS		Mac OS				
	$\hat{\beta}$	RSD	$\hat{\beta}$	RSD	$\hat{\beta}$	RSD	$\hat{\beta}$	RSD	$\hat{\beta}$	RSD			
Filip	7.1	8.2	-	1.1	0	1.1	0	1.1	0	0	0	0	0
Longley	-	0	-	-	0	-	0	-	0	-	0	-	0
Norris	13.4	14.1	-	1.8	12.1	1.8	10.0	12.3	1.8	10.7	1.6	10.0	1.6
Noint1	0	0	Inf	2.9	Inf	2.9	0	Inf	2.9	0	0	0	0
Noint2	0	0	Inf	2.3	Inf	2.3	0	Inf	2.3	0	0	0	0
Pontius	3.6	13.2	6.4	1.6	7.1	1.6	5.5	6.2	1.6	5.3	1.4	5.5	1.4
Wampler1	9.3	9.4	6.9	4.6	7.0	4.8	2.9	6.6	4.4	3.6	3.6	4.1	4.1
Wampler2	Inf	14.2	9.6	9.3	9.8	9.5	6.9	9.8	0	8.5	8.5	6.0	8.3
Wampler3	-	1.6	6.9	0	7.0	0	2.9	6.6	0	2.9	0	3.5	0
Wampler4	-	1.6	6.9	0	7.0	0	2.9	6.6	0	2.9	0	3.5	0
Wampler5	-	1.6	6.5	0	7.0	0	2.9	6.2	0	2.9	0	3.5	0

LRE of those eigenvalues that should take value m (there are $n - 1$ of them) denoted ℓ_m , and (vi,vii) the percentage of eigenvalues which test equal to m and to n (being the correct answers $n - 1$ and $m - 1$, respectively) denoted ℓ_N and ℓ_M . Those results are presented in tables 12 (for Matlab), 13 (for Octave) and 14 (for SciLab).

4 CONCLUSIONS

Regarding the computation of basic statistics, Table 1 shows that the mean poses little difficulty for the platforms, with the exception of Octave for Linux, which presented the smallest number of LRE in five of the nine datasets. Surprisingly, these five datasets offer low numerical difficulty. The same platform performs the worst also when computing the standard deviation, but only in five out of nine cases, while SciLab and Matlab presented an unacceptable low accuracy in a single dataset each. As in other studies, c.f., Almiron et al. (2010), the first-lag sample autocorrelation is a challenging quantity to compute. The three best results, which are still unacceptable, were computed by Matlab (PiDigits) and Octave under Windows (PiDigits and Michelson). Though SciLab had the overall worst performance, none of the platforms performed well.

SciLab presented the best performance when dealing with the binomial, t-Student and F distributions, and also when computing the cumulative distribution function of the Poisson law. Octave fails to produce acceptable values when dealing with the binomial, t-Student and F distributions, and also when computing the cumulative distribution function of the Poisson law, but it is the best one dealing with the normal distribution and pairs the other platforms with the gamma law.

Matlab failed at computing the t-Student distribution; in every assessed case, it returned an error message. This is a serious issue due to the widely spread use of this distribution in basic statistical tests.

Though a single platform among the ones here assessed is unable to provide consistently good answers, SciLab is the one that presented the best overall performance.

Six out of eleven linear regression datasets were not dealt with any of the considered platforms in an adequate manner. Only Matlab provided acceptable results for `Filip` and for `Wampler1`. `Wampler1` and `Wampler2` were acceptably treated by Matlab, Octave under Windows and SciLab, while only Octave under Linux returned useful values for `Pontius`. Again, no single platform can be advised as safe for the linear regression problems here considered.

The best results were provided by Matlab, SciLab under Windows and under Mac OS, and by Octave under Mac OS when making decisions about the determinant of ill-conditioned matrices. Octave under Windows and Linux, and SciLab under Linux provided an unacceptable number of erroneous results. Users are advised to be very careful when testing equality between a value of interest and a numerical computation involving determinants in these platforms.

The assessment based on spectral graph analysis presented a very consistent behavior with respect to the problem size (the bigger the graph, the worse the answer), being ℓ_M and ℓ_N the most sensitive quantities across all platforms and operating systems, and they can be reported as good in most cases. The first and last eigenvalues (ℓ_1 and ℓ_{m+n}) are always dependable if computed in double precision and then tested in single precision, being the latter consistently more precise than the former. The balance of bipartite connected graphs did not have a strong impact on the results, except for the percentage of correct eigenvalues.

Extreme care must be taken when making decisions about graphs based on their spectral properties. As a rule of the thumb, double-precision computation is advised, but the com-

parison to known values should be made rounding or, at most, using at most floating point representation.

Regarding the variability among operating systems, SciLab was more consistent than Octave in most of the situations under assessment.

REFERENCES

- Almiron M., Almeida E.S., and Miranda M. The reliability of statistical functions in four software packages freely used in numerical computation. *Brazilian Journal of Probability and Statistics*, Special Issue on Statistical Image and Signal Processing:107–119, 2009. doi: 10.1214/08-BJPS017.
- Almiron M., Vieira B.L., Oliveira A.L.C., Medeiros A.C., and Frery A.C. On the numerical accuracy of spreadsheets. *Journal of Statistical Software*, 34(4):1–29, 2010.
- Bollobas B. *Modern Graph Theory*. Springer, 1998.
- Fiedler M. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.
- Ho N. and Van Dooren P. On the pseudo-inverse of the Laplacian of a bipartite graph. *Applied Mathematics Letters*, 18(8):917–922, 2005. doi:10.1016/j.aml.2004.07.034.
- Knüsel L. On the accuracy of statistical distributions in Microsoft Excel 2003. *Computational Statistics & Data Analysis*, 48:445–449, 2005.
- Kruck S.E. Testing spreadsheet accuracy theory. *Information and Software Technology*, 48(3):204–213, 2006. ISSN 0950-5849. doi:10.1016/j.infsof.2005.04.005.
- McCullough B.D. and Heiser D.A. On the accuracy of statistical procedures in Microsoft Excel 2007. *Computational Statistics & Data Analysis*, 52(15):4570–4578, 2008. doi:10.1016/j.csda.2008.03.004.
- National Institute of Standards and Technology. Statistical reference datasets: Archives. 1999. Last checked on January 31, 2010.
- Yalta A.T. The accuracy of statistical distributions in Microsoft©Excel 2007. *Computational Statistics & Data Analysis*, 52:4579–4586, 2008. doi:10.1016/j.csda.2008.03.005.

Table 12: Accuracy of MatLab computing spectral graph analyses under Windows

m, n	ℓ_1	ℓ_{m+n}	ℓ_S	ℓ_n	ℓ_m	ℓ_N	ℓ_M
9, 10	15.2	Inf	15.8	14.9	14.9	11.1	62.5
99, 100	12.7	14.8	Inf	14.6	13.8	7.1	4.1
999, 1000	10.5	14.5	15.1	13.8	12.6	1.3	2.3
2, 17	15.3	Inf	15.7	Inf	14.7	12.1	100.0
2, 197	14.3	15.8	15.2	inf	11.7	6.1	100.0
2, 1997	13.1	15.9	14.2	15.9	9.6	1.1	0

Table 13: Accuracy of Octave computing spectral graph analyses

m, n	Windows										Mac OS										
	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M
9, 10	Inf	Inf	15.8	15.3	15.1	0	11.1	15.3	Inf	15.8	15.5	15.0	25.0	44.4	15.3	Inf	Inf	14.8	15.2	12.5	22.2
99, 100	13.7	15.9	Inf	14.9	14.8	9.2	11.1	13.6	15.9	Inf	14.8	14.6	4.1	5.1	13.6	15.9	15.4	13.9	14.3	6.1	5.1
999, 1000	11.9	14.9	Inf	13.7	13.2	3.4	2.0	12.1	15.5	15.5	13.7	13.4	1.4	1.8	10.8	14.2	14.8	12.8	13.1	2	0
2, 17	15.4	Inf	15.7	Inf	15.0	100.0	37.5	14.8	15.4	Inf	15.7	14.8	0	12.5	15.5	Inf	15.7	14.4	15.4	0	100.0
2, 197	14.6	15.9	Inf	15.5	13.3	0	11.2	Inf	15.9	15.8	15.8	12.5	0	15.8	14.8	15.2	14.3	12.3	14.0	0	5.6
2, 1997	14.4	15.9	15.9	15.9	10.7	0	8.6	14.1	15.6	15.3	15.6	10.2	0	11.7	13.1	14.9	14.5	9.6	14.5	0	2.2

Table 14: Accuracy of SciLab computing spectral graph analyses

m, n	Windows										Linux										Mac OS														
	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M	l_1	l_{m+n}	l_S	l_n	l_m	l_N	l_M
9, 10	14.6	Inf	15.8	15.1	14.9	37.5	44.4	15.2	Inf	15.8	15.4	15.0	25.0	44.0	15.2	Inf	Inf	15.1	14.8	12.5	44.0	15.2	Inf	Inf	15.7	14.2	14.8	12.5	22.2	Inf	15.7	15.4	14.4	0	37.5
99, 100	13.4	15.5	15.7	14.6	14.2	5.1	10.1	14.3	15.5	14.8	14.7	Inf	7.1	6.1	13.1	15.4	15.7	14.2	13.8	4.1	5.0	13.1	15.4	15.7	14.2	13.8	4.1	5.0	13.1	15.4	15.7	14.2	13.8	4.1	5.0
999, 1000	12.1	14.9	15.6	13.6	13.3	1.1	3.9	11.4	14.9	15.6	13.6	14.1	1.0	1.8	10.9	14.2	14.8	13.3	13.1	1.3	1.1	10.9	14.2	14.8	13.3	13.1	1.3	1.1	10.9	14.2	14.8	13.3	13.1	1.3	1.1
2, 17	14.8	15.4	Inf	Inf	14.9	100.0	12.5	14.8	14.8	15.7	15.4	Inf	0	12.5	15.5	Inf	15.7	15.4	14.4	0	37.5	15.5	Inf	15.7	15.4	14.4	0	37.5	15.5	Inf	15.7	15.4	14.4	0	37.5
2, 197	14.5	15.8	15.1	15.0	12.8	0	12.2	14.5	15.8	Inf	15.4	13.0	0	28.1	14.2	15.6	14.3	14.0	12.0	0	17.3	14.2	15.6	14.3	14.0	12.0	0	17.3	14.2	15.6	14.3	14.0	12.0	0	17.3
2, 1997	13.0	15.6	14.1	13.5	11.5	0	6.0	13.0	15.6	Inf	14.0	10.7	0	19.4	13.0	15.6	13.5	13.0	10.0	0	8.1	13.0	15.6	13.5	13.0	10.0	0	8.1	13.0	15.6	13.5	13.0	10.0	0	8.1