On the Importance of Verifying Forecasting Results

A. Talha YALTA & Olaf JENAL

Department of Economics
TOBB University of Economics and Technology

Working Paper No: 08-04
TOBB University of Economics and Technology
Department of Economics

June 2008

ON THE IMPORTANCE OF VERIFYING FORECASTING RESULTS

A. Talha Yalta[*] Olaf Jenal[†]

February, 2008

Abstract

We discuss the various sources of error in numerical computations with the use of examples from the literature relevant to time series analysis. We also submit a case where, by manual verification, we were able to discover a plausible forecast to be erroneous due to a number of software flaws in the XLSTAT addin for Microsoft Excel. Furthermore, after discussing the alternative techniques for implementing on a computer the ARIMA (AutoRegressive, Integrated, Moving Average) methodology, we show that different approaches can cause considerable discrepancies in the results across different programs and even within a single software system.

1 Introduction

The use of advanced time-series models and complex forecasting algorithms via various commercial software packages allows improved forecast performance by the practitioners (Sanders and Manrodt, 2003). On the other hand, the accuracy of software cannot be taken for granted and it is possible to find serious errors in commonly used statistical programs as shown by surveys such as McCullough (1999), McCullough (2004), and Keeling and Pavur (2007). It is also documented by Newbold et al. (1994), McCullough and Vinod (2003), and Stokes (2004) that dissimilar sets of results can be obtained when the same data and model is analyzed through different econometric programs. It is evident that researchers and practitioners should treat computer output with a degree of circumspection and verify computer generated forecasts manually or by using more than one software package where available. This practice not only allows fewer errors and more accurate forecasts, but also leads to better programs because developers do correct errors discovered by users as shown by Keeling and Pavur (2007) and McKenzie and Takaoka (2007).

Our main objective in this study is to draw attention to the sources of numerical failures that can be encountered in commonly used forecasting programs. We also demonstrate the pitfalls of uncritically accepting output from a single software package by reporting on

[*] Correspondence to: TOBB University of Economics and Technology; Sogutozu Caddesi No:43; Sogutozu, 06560, Ankara, Turkey E-mail: yalta@etu.edu.tr

[†] trans-o-flex Schnell-Lieferdienst GmbH & Co. KG, Weinheim, Germany

our attempts to correctly estimate a seasonal ARIMA (AutoRegressive, Integrated, Moving Average) model by using the XLSTAT data analysis addin for Microsoft Excel. An important third goal is to show that various add-on modules, which claim to provide more and better functionality to programs such as Excel, can also have errors of their own and thus need to be evaluated thoroughly with the usual[1] introductory and intermediate tests of software reliability.

In the next section, we briefly discuss the various types of errors in numerical computations and the consequences of ignoring computational realities with the use of examples from the literature relevant to time series analysis and forecasting. In section 3, we document a case where it was possible by hand computation to find what appeared to be a plausible computer forcast to be erroneous due to a number of software flaws. This is followed by a discussion of our attempts to verify the estimation results using alternative software packages. Some conclusions are drawn in Section 5.

2 Sources of Software Failures

The practice of forecasting often involves complex and computationally intensive analysis. On the other hand, many resources on this topic give the impression that all one has to do is to use a computer to apply various techniques. It is long known, however, that a forecaster who estimates a model with two different programs can obtain two different sets of results. Indeed, Küsters et al. (2006) mentions from the period of 1970s an industrial practitioner, who would run the same model on four different mainframe packages and accept a forecast if the same answer is given by at least three out of the four programs. This example shows that the problem of errors and discrepancies in computer generated results is an ongoing concern in the forecasting community for at least thirty years.

The accuracy of software cannot be assumed. Today, most econometric and forecasting programs perform calculations and store the results using double precision floating-point numbers. However, this numerical system cannot completely mimic the real numbers and can result in computational mistakes, namely rounding errors, cancellation errors, overflow errors, and truncation errors. These, together with human errors such as algorithm errors and implementation errors make it crucial for the users of forecasting software to have an understanding of the computer arithmetic and software limitations before engaging in advanced data analysis.

Perhaps the most common and critical accuracy problem regarding computer based calculations is the rounding error, which is due to the hardware limitation that certain numbers can not be fully represented by computers. For example, the computer representation of the decimal 0.1 is the repeating binary fraction $0.00011\overline{0011}$, which becomes 0.10000000149011612 when converted back to decimal with single precision, the default data storage method in many software packages. Cancellation error is a special case of the rounding error, which happens after subtraction of two nearly equal numbers, leaving only the accumulated rounding error as a result. One example of a method extremely susceptible to rounding and cancel-

[1] The "Statistics Quiz" discussed in detail by Sawitzki (1994) and the methodology proposed by McCullough (1998b) are the two widely applied tests for assessing accuracy of programs offering statistical functionality.

lation errors is the Yule-Walker equations used for computing the partial autocorrelations in the Box-Jenkins modeling of time series data. Perhaps due to historical reasons, many forecasting packages continue to employ this method, although it is known (see McCullough, 1998a) to produce results inferior to those obtained using alternative procedures.

A related problem is the overflow and underflow errors, which happen when computations are done in such a way that intermediate calculations exceed the range of values capable of being represented by the computer, which are typically between $[4.9 \times 10^{-324}, 1.8 \times 10^{308}]$ as the smallest and largest positive numbers respectively. Once encountered, an underflow or an overflow can generate a hardware interrupt, set a status bit or, in many cases, just be ignored by the program, resulting in a misleading answer. A good example to this type of error is provided by Zeileis and Kleiber (2005), which discusses an underflow with no easy workaround in GAUSS version 3.2.32, which rendered invalid the original results of the multiple structural change model proposed by Bai and Perron (2003).

Unlike the rounding and overflow errors, which are due to the limitations of the hardware, the truncation error is caused by the limitations of the software. Nonlinear methods such as GMM, GARCH, or seasonal ARIMA models theoretically involve an infinite number of iterations. The computer estimation, however, can include only a finite number of calculations resulting in a truncation error when the operation is terminated after a given number of iterations or when the relative change in either the objective function or the estimates is smaller than a predetermined convergence criterion.

Algorithm errors arise from the fact that there is more than one way to solve the same problem, some better than others. For example, correction for AR(1) first order autocorrelation can be performed using various different medhods including Cochran-Orcutt, Hildreth-Lu, Beach-MacKinnon, and Prais-Winsten. These methods employ dissimilar algorithms and objective functions[2] and therefore can return different answers as demonstrated by Lovell and Selover (1994).

A fifth type of accuracy errors are implementation errors, which refers to the failure to program the computer to exactly follow the operations specified by a particular algorithm claimed to be used in the software. An example is the various bugs that we discuss in this paper, which result in the XLSTAT program failing to estimate correctly a seasonal ARMA model as well as computing a forecast accurately.

Computer arithmetic is completely different than pencil-and-paper mathematics and research results obtained using computers are sensitive to the choice of both software and hardware. A common belief is that since many data sets are accurate to only a few digits, a high level of accuracy is not needed. This is not true because there is a dichotomy between the use of output and the calculation of output. As McCullough and Vinod (1999) argues, while reporting 10 digits of a solution is not necessary, all intermediate calculations should be done with as many digits as possible. What makes floating point calculations problematic is how small errors can accumulate after successive operations. Many techniques such as

[2] As the objective function, Cochran-Orcutt and Hildreth-Lu use the conditional maximum likelihood method while Beach-MacKinnon and Prais-Winsten employ the exact maximum likelihood, and the generalized least squares (GLS) respectively. For conditional maximum likelihood estimation, Cochran-Orcutt is known to be relatively easier to implement and it finds a local optimum while Hildreth-Lu is fairly easy to implement and finds a global optimum. Prais-Winsten is often preferred over Beach-MacKinnon, which requires the additional assumption that variables are normally distributed.

simulations and nonlinear estimation involve operations carried out an enormous number of times which makes it perfectly possible that the final result will be accurate to only one or even zero digits if the necessary attention is not given to the computational aspect of solving the econometric problem. For further information about errors in numerical computations, the reader is referred to a source such as Altman et al. (2004), which provide a more detailed account on this subject.

3 Manual verification of an XLSTAT ARIMA model

Our experience with the XLSTAT program can help demonstrate the dangers of ignoring the computational realities and uncritically accepting results computed by a single software system. Offered by Addinsoft, XLSTAT is a popular add-on program enhancing the analytical capabilities of Microsoft Excel with custom developed software components for regression, data analysis, visualization, and forecasting. XLSTAT offers over 100 statistical procedures integrated into Excel and Addinsoft claims *without offering any evidence* that "the quality of the computations (carried out by the program) is identical to that offered by the leading scientific packages." Addinsoft also claims *without providing any proof* that "all XLSTAT-Time (component) functions have been intensively tested against other software to guarantee the users fully reliable results."[3]

We had a chance to test Addinsoft's claims while working on a forecast project for a logistics services company near Mannheim in Germany. Our project involved running XLSTAT for daily forecasting of quantities entering the company network at various specific depots while maintaining a horizon of one week into the future. In order to get acquainted with the program, we attempted to fit to our data several ARIMA models including a seasonal ARIMA$(1,0,0)(1,1,0)_5$ model[4] of the mathematical form

$$(1 - L^5)Y_t = \alpha + \frac{1}{(1 - \phi_1 L)(1 - \phi_{s,1} L^5)} u_t \tag{1}$$

where Y_t is the dependent variable, u_t is the error term, L is the lag operator, α is the constant, and ϕ_1 and $\phi_{s,1}$ are the nonseasonal and seasonal autoregressive coefficients respectively.

The estimation of the above model using XLSTAT involves only a few mouse clicks and takes less than a second. On the other hand, we know that calculations on computers can involve errors that can in turn lead to bad analytics. As a result, we decided to hand replicate the XLSTAT forecasts using the forecast equation and the estimated parameters for α, ϕ_1, and $\phi_{s,1}$. Because this is a fairly simple calculation involving a weighted sum of several periods and a constant, little did we know that it was going to take us four attempts to obtain a consistent set of forecasts from XLSTAT.

In our first attempt, we failed to reproduce the results of XLSTAT using the coefficient estimates of XLSTAT. As Table 1 shows, our manually computed forecasts and those given

[3] See the XLSTAT product webpages at `http://www.xlstat.com/en/products/` and `http://www.xlstat.com/en/products/xlstat-time/` respectively. (accessed September 5, 2007)

[4] We initially chose this model randomly and stumbled onto the programming errors. When we went back to try a better fitting model, it would not run in the original version of XLSTAT!

Table 1: Parameter Estimates and Forecasts by XLSTAT 2007

	Attempt 1		Attempt 2		Attempt 3		Attempt 4	
Parameters	(xlstat 2007)		(+patch 1)		(+custom α)		(+patch 2)	
α	-1967.42		-1566.54					
ϕ_1	0.365		0.490					
$\phi_{s,1}$	-0.181		-0.378					
Forecasts	xlstat	manual	xlstat	manual	xlstat	manual	xlstat	manual
$t+1$	22290.5	22192.3	23416.6	22753.2	correct	23416.6	correct	22753.2
$t+2$	21781.6	21553.4	22861.3	22197.8	22861.3	23186.4	correct	22197.8
$t+3$	20643.2	20307.8	21079.4	20415.9	21079.4	21563.8	correct	20415.9
$t+4$	19252.4	19184.7	19878.2	19214.7	19878.2	20440.6	correct	19214.7
$t+5$	17007.9	17045.2	17846.6	17183.2	17846.6	18447.3	correct	17183.2

by the program are slightly but noticeably different from each other. We decided to report the problem to Addinsoft's support department, who responded to our inquiry kindly and quickly. Indeed, within a few days, we were told that the problem has been solved and a corrected version of the program was now available.

Using the updated package, in our second attempt, we discovered that the XLSTAT estimates after the first patch were quite different. As Table 1 shows, the discrepancy in the $\phi_{s,1}$ parameter in particular is more than 100%. We also discovered that the forecasts computed with the "corrected" version, while significantly different than the earlier output, continued to disagree with our hand calculations. Once informed about the continued discrepancy with our own results, the software vendor claimed that the output given by XLSTAT was the correct forecast. Further, we were advised that using the constant term reported by the program in our own computations was inappropriate since this was the constant "for the prediction for the differenced series." Consequently, according to the support team, what needed to be done in order for us to be able to reproduce the forecasts manually was to use in our formula the constant $k = 2c - m$, where c is the constant reported by XLSTAT and m is the mean of the differenced series.

Following the the new directions, we gained partial success in our third attempt. That is, we were able to replicate the forecast given by the program for the initial period only. Moving on to the one week horizon, however, we realized that modifying the suggested formula for $(t + 2, \ldots, t + 5)$ yielded forecasts again different than our hand calculations. Contacting Addinsoft regarding the persisting inconsistencies resulted in their identification of another problem and soon they supplied us a new patch also acknowledging that our initial approach of using the constant displayed by XLSTAT was in fact correct.

Attempting for the fourth time using the second patch, XLSTAT forecasts finally agreed with our manual computations. As can be seen from Table 1, there are noticeable differences in the forecasts over the four attempts and the consistency between our manually computed forecasts and those given by XLSTAT can be as low as zero significant digits. We would not have discovered these errors if we did not have more trust in our hand computations than in XLSTAT, what one might consider to be a sophisticated tool for forecasting.

It is important to note at this point that, with the two patches released in response our inquiries, Addinsoft merely claimed to correct the errors without providing a proof,

Table 2: XLSTAT Forecasts for Several Classical Series

Series	Model	Forecasts	xlstat 2007	+ patch1	+ patch2
A	(1,0,1)	t+1	17.06	1.87	17.38
		t+2	17.06	1.84	17.35
		t+3	17.06	1.82	17.32
B	(0,1,1)	t+1	357.00	357.38	357.38
		t+2	357.00	357.38	357.38
		t+3	357.00	357.38	357.38
C	(1,1,0)	t+1	18.64	18.64	18.64
		t+2	18.50	18.50	18.50
		t+3	18.39	18.39	18.39
D	(1,0,0)	t+1	9.10	1.17	9.10
		t+2	9.11	1.18	9.11
		t+3	9.11	1.18	9.11
E	(2,0,0)	t+1	92.11	59.52	92.11
		t+2	91.23	58.64	91.23
		t+3	77.06	44.47	77.06
F	(2,0,0)	t+1	61.23	69.02	61.23
		t+2	42.41	50.20	42.41
		t+3	55.99	63.78	55.99

a benchmark result or a comparison to another package. Without these, there is enough reason[5] for the user not to assume that an update offered by the software vendor will correct a software flaw properly and without introducing a new flaw. We attempted to test this by fitting several classical Box-Jenkins (1970) series with the “fixed” versions of XLSTAT. As Table 2 shows, the first patch indeed brings a new flaw, which can lead to important forecasts errors. This vanilla bug, which somehow does not affect Box-Jenkins’ series B and C and our initial model, must have been discovered and fixed with the second patch. Conveniently, the software vendor neglected to mention to us about this issue. In fact, in the following months, Addinsoft did not announce to the other users of the program any of the errors or the subsequent patches discussed in this paper. We were, however, offered a 50% discount on a one user license of XLSTAT-Pro+Time, which we did not take advantage of.

Finally, it is worth mentioning about yet another bug that we discovered during our attempts to obtain consistent forecasts using XLSTAT. For ARIMA estimation, XLSTAT has two options namely “likelihood” and “least-squares”. According to the program’s online

[5] See, for example, McCullough (2008), who discuss how Microsoft twice wrongly claimed to fix the bad random number generator in Excel by implementing the Wichmann and Hill (1982) RNG. It is shown that Excel 2007 still uses “an unknown and undocumented RNG of unknown period that is not known to pass any standard tests of randomness”.

Table 3: XLSTAT Estimates for Several Classical Series

Series	Model	Param.	xlstat (ml)	xlstat (ls)	Box-Jenkins
A	(1,0,1)	α	1.56	3.57	1.45
		ϕ_1	0.91	0.79	0.92
		θ_1	-0.58	153.69	-0.58
B	(0,1,1)	θ_1	0.09	-44647194.13	0.09
C	(1,1,0)	ϕ_1	0.82	0.82	0.82
D	(1,0,0)	α	1.20	1.17	1.17
		ϕ_1	0.87	0.87	0.87
E	(2,0,0)	α	14.34	14.35	14.35
		ϕ_1	1.41	1.42	1.42
		ϕ_2	-0.71	-0.73	-0.73
F	(2,0,0)	α	58.92	58.88	58.87
		ϕ_1	-0.34	-0.35	-0.34
		ϕ_2	0.19	0.19	0.19

help file, the likelihood option "maximize(s) the likelihood of the parameters knowing the data", whereas the least-squares option "minimize(s) the sum of squares of the residuals". Because the two estimation methods optimize the same objective function and involve a similar set of calculations, one would expect the two options to yield the same results. However, we have noticed that XLSTAT estimates were noticeably different depending on the choice of this parameter. In order to further investigate this discrepancy, we again decided to fit several classical Box-Jenkins series using the two options. Table 3 shows the XLSTAT estimates for our model along with the results for several benchmark models reported by Box and Jenkins (1970). Clearly, the least-squares estimation option in XLSTAT can produce θ moving average estimates that are grossly erroneous. Once notified about this problem, Addinsoft acknowledged that they knew "the least-squares method is not reliable in some cases" and that they were considering to remove this option in the future. It is incredible that Addinsoft knew the least squares option is not reliable, but did not warn the users! *They allowed users to use this function thinking it was correct!* The faulty least-squares estimation function was still available at the time of writing of this article.

4 Further verification using alternative packages

Being able after four attempts to obtain a consistent (albeit more than one) set of answers from XLSTAT, we decided to see if other time-series analysis software packages also have difficulty finding accurate estimates as well as forecasts for the computation of our seasonal ARIMA$(1,0,0)(1,1,0)_5$ model. We turned our attention to four additional programs namely

Autobox, GRETL, RATS, and X-12-ARIMA.

There are three main approaches to ARIMA estimation namely the unconditional maximum likelihood method, the conditional maximum likelihood method and backcasting[6]. The unconditional maximum likelihood approach, also known as exact maximum likelihood, generates given a sample size of T a full T×T covariance matrix, for which the log likelihood can be computed efficiently[7] by using the Kalman filter. Nearly identical is the "unconditional least squares method", which, instead of maximizing the unconditional log likelihood function, minimizes the unconditional error sum of squares. The second main approach is the conditional maximum likelihood (conditional least squares) procedure, which has the objective of maximizing the likelihood (minimizing the sum of squared errors) conditional on the first observations. This method involves an iterative least squares procedure since the residuals are a non-linear function of the observables when there are multiplicative autoregressives or moving average terms in the model. The third main approach is backcasting (or backforecasting), which was initially proposed by Box and Jenkins (1970) as a computationally convenient approximation to the exact maximum likelihood method. The idea behind backcasting is that, for a univariate model, the forward and backward representations of a stationary ARMA model are the same. Consequently, the expected value of pre-sample data can be approximated by starting from the end of the data set, recursing back toward the beginning, and then "back forecasting" into the pre-sample period. The resulting inaccuracies will be relatively small provided that the backcasts are far enough before the actual data. These three methods are asymptotically equivalent under the standard assumptions, however, because they involve dissimilar objective functions, in practice they often result in unequal coefficient estimates as shown by Newbold et al. (1994).

The three approaches discussed above have various advantages and shortcomings in comparison to each other. One advantage of the exact maximum likelihood approach is that it produces results that are directly comparable across different software systems. However, it can become ill-behaved for a large dimensional specification such as an ARMA(6,6) model. Conditional maximum likelihood does not have this problem but it can give different results depending on the method chosen for generating the pre-sample moving averages. The problem with the backcasting method, on the other hand, is that the estimator depends on the number of the backcast periods being used. Also, it does not easily generalize to any model with intervention terms or other exogenous variables. As a result, with today's computing power, the practical importance of backcasting has become negligible, although it still remains in some packages as a legacy of earlier programming.

Aside from the various objective functions that can be used, another source of discrepancy in the estimates of a Box-Jenkins model across different programs is the parameterization of the intercept term. In ARIMA methodology, there are several ways of handling the nonzero

[6] In addition to these three methods, ARIMA models including only AR terms can be fitted by ordinary least squares and a seasonal-multiplicative model can be estimated using the nonlinear least squares procedure. We consider these two additional methods to be outside the scope of this study.

[7] That is, by avoiding inversion of the T×T matrix. Matrix inversion is a computationally intensive procedure. For example, inverting a 100×100 matrix involves a third of a million operations, which can result in significant accumulated rounding errors especially when the elements of the matrix differ in size (Stokes, 2005). The Kalman filter, which involves inversions of the much smaller covariance matrices, is not as demanding computationally.

mean α which include the unfiltered constant

$$\phi(L)\phi_s(L^s)(1-L)^d(1-L^s)^D Y_t = \alpha_1 + \theta(L)\theta_s(L^s)u_t, \tag{2}$$

the filtered constant term as in

$$(1-L)^d(1-L^s)^D Y_t = \alpha_2 + \frac{\theta(L)\theta_s(L^s)}{\phi(L)\phi_s(L^s)}u_t, \tag{3}$$

as well as the rather nonstandard

$$\Big[(1-L)^d(1-L^s)^D Y_t - \alpha_3\Big] = \frac{\theta(L)\theta_s(L^s)}{\phi(L)\phi_s(L^s)}u_t \tag{4}$$

where $\phi(L)$ and $\theta(L)$ are the autoregressive and moving-average operators, and $\phi_s(L^s)$ and $\theta_s(L^s)$ are the seasonal autoregressive and seasonal moving-average operators all represented as a polynomial in the lag operator respectively.

The first definition above estimates the intercept as part of a regression model on the differenced data, leading more directly to a forecasting model, while the second definition treats the constant as a mean and estimates it jointly with the other parameters. One can easily transform between α_1 and α_2 since they are directly related by simple algebra such that $\alpha_1 = \phi(L)\phi_s(L^s)\alpha_2$. The two models yield the same forecasts regardless of the notation used for the intercept, however, the fact that the reported estimates of the intercept can be substantially different depending on which package is used can be a source of considerable confusion for the practitioner.

Equations (3) and (4) are equal mathematically but different computationally. Unlike the first and the second definitions, the third definition implies a model without the intercept term. It involves first the computation of the mean of the differenced data which is then subtracted off, leading to the OLS estimate of the mean, estimation of which along with the other parameters, as in (3), yielding the GLS estimates. Because of the difference in the computational approach, α_1 and α_3 are not numerically identical and there can be slight differences in the ARMA parameter estimates as well.

In addition to the various objective functions and the alternative methods for modeling the intercept, a third potential source of discrepancy in the parameter estimates of a Box-Jenkins model is the choice of the minimization algorithm used for solving the nonlinear optimization problem. Among the numerous alternatives, some frequently employed in econometric software are Gauss-Newton, Levenberg-Marquardt (Marquardt, 1963), BFGS (Broyden-Fletcher-Goldfarb-Shanno), and BHHH (Berndt, Hall, Hall, and Hausman, 1974). These algorithms belong to a class of hill-climbing optimization techniques that seeks the local optimum uphill from the initialization points. Another commonly used method is the AS 197 algorithm by Mélard (1984), which provide a fast and memory efficient method to compute the exact maximum likelihood function of a stationary ARMA process of order (p, q). There is also the simplex and genetic search methods, which are derivative-free methods that, unlike the above, do not require the formula be twice continuously differentiable, however, they also cannot compute standard errors. See Avriel (2003) for information regarding various points that need to be considered when choosing between the various minimization algorithms.

Table 4: Exact ML Estimates and Forecasts by Four Programs

Parameters	XLSTAT 2007	XLSTAT (+patch 2)	GRETL 1.7.3	RATS 7.0	X-12-ARIMA 0.3
α_2			-2284.68	-2284.68	-2284.68
α_3	-1967.42	-1566.54			
ϕ_1	0.365	0.490	0.489	0.489	0.489
$\phi_{s,1}$	-0.181	-0.378	-0.378	-0.378	-0.378
Forecasts					
$t+1$	22290.5	22753.2	22712.9	22712.9	22712.9
$t+2$	21781.6	22197.8	22138.8	22138.8	22138.8
$t+3$	20643.2	20415.9	20346.9	20346.9	20346.9
$t+4$	19252.4	19214.8	19141.5	19141.5	19141.5
$t+5$	17007.9	17183.2	17108.5	17108.5	17108.5

Note: α_1 for RATS, GRETL and X-12-ARIMA is computed as -1607.74.

According to their respective user's guides and online documentations[8], Autobox, GRETL, RATS, X-12-ARIMA, and XLSTAT all offer different combinations of objective functions, constant terms, as well as minimization algorithms for ARIMA estimation. For the objective function, Autobox employs the conditional least squares approach, while XLSTAT offers unconditional least squares and unconditional maximum likelihood estimation. The RATS function **boxjenk** provides the unconditional least squares and the conditional maximum likelihood options, while the GRETL function **arima**, and the X-12-ARIMA function **estimate** both offer the exact and the conditional maximum likelihood options. For the intercept, RATS and X-12-ARIMA by default employ the filtered constant term α_2, while Autobox and XLSTAT use α_1 and α_3 respectively. GRETL uses α_1 for the conditional maximum likelihood method and α_2 for the exact maximum likelihood method. Finally, for the minimization algorithm, both Autobox and X-12-ARIMA employ the Levenberg-Marquardt algorithm[9], while XLSTAT and GRETL use the Mélard's and the BFGS methods respectively. RATS offers a choice among Gauss-Newton, BFGS, simplex, and the genetic search algorithms.

We compared the output of the five programs by computing the five period forecasts after fitting to the data our model using where available both the exact maximum likelihood and the conditional maximum likelihood methods with the default option for the computation of the intercept as well as the minimization algorithm. In all estimations, the convergence criterion was chosen so that each solver produces a stable answer. The computations were carried out using an Intel Centrino Duo 2.16GHz notebook computer with 2GB memory. The operating system used was Microsoft® Windows XP.

[8] See Automatic Forecasting Systems (2007), Cottrell and Lucchetti (2008), Estima (2007), U.S. Census Bureau (2007), and Addinsoft (2007).

[9] X-12-ARIMA uses the Levenberg-Marquardt algorithm from the free MINPACK FORTRAN library, while Autobox employs its custom implementation of this method with proprietary speed and efficiency improvements for time series analysis.

Table 5: Conditional ML Estimates and Forecasts by Four Programs

Parameters	Autobox 6.0	GRETL 1.7.3	RATS 7.0	X-12-ARIMA 0.3
α_1	1786.24	-1786.24		
α_2			-2615.24	-2615.24
ϕ_1	0.512	0.512	0.512	0.512
$\phi_{s,1}$	-0.401	-0.401	-0.401	-0.401
Forecasts				
$t+1$	22545.4	22545.4	22545.4	22545.4
$t+2$	21857.3	21857.3	21857.3	21857.3
$t+3$	19934.2	19934.2	19934.2	19934.2
$t+4$	18680.2	18680.2	18680.2	18680.2
$t+5$	16636.7	16636.7	16636.7	16636.7

Note: α_1 for RATS and X-12-ARIMA is computed as -1786.24.

Table 4 and Table 5 show the parameter estimates and the forecasts obtained from alternative programs for the exact ML and conditional ML methods respectively. Autobox, GRETL, RATS, and X-12-ARIMA are all in perfect agreement on ϕ_1, $\phi_{s,1}$, as well as the constant term, modulo the decision to print either the unfiltered or the filtered intercept, which in this case differ from each other by a factor of $(1-\phi_1) \times (1-\phi_{s,1})$. The XLSTAT parameter estimates and forecasts after the second patch are similar but noticeably different in comparison to those reported by GRETL, RATS, and X-12-ARIMA, which is expected due to the dissimilar computational approach for modelling the constant term. The two tables reveal that the rule-of-thumb method of accepting a forecast if given by at least three out of four software packages, employed by the industrial practitioner from the 1970's, might indeed be a useful practice after all.

Finally, in Table 6 we present a comparison of the BFGS, Gauss-Newton, simplex, and genetic minimization algorithms used in RATS jointly with the exact maximum likelihood method. The table shows that BFGS has difficulty to converge when the tolerance is large whereas Gauss-Newton and genetic has difficulty to converge when the tolerance is small. Simplex and genetic are slower to converge and can be inaccurate for larger tolerance levels. The parameter estimates in general are similar although, depending on the model and the dataset, there can be significant dissimilarities in this department as well, as shown by Newbold et al. (1994).

5 Conclusions

Forecasting software is advancing at a steady rate and with every new version of different programs comes new functionality. Thanks to the availability of advanced software tools, today users are not necessarily required to be specialized in econometrics and forecasting in order to “analyze” data. Sanders and Manrodt (2003) shows that the majority of the

Table 6: Comparisons of the Results Using Different Minimization Algorithms

	BFGS				Gauss-Newton			
tolerance	conv.	α_2	ϕ_1	$\phi_{s,1}$	conv.	α_2	ϕ_1	$\phi_{s,1}$
1.00E-02	nc				7	-2284.69	0.489	-0.378
1.00E-04	13	-2284.68	0.489	-0.378	7	-2284.69	0.489	-0.378
1.00E-06	15	-2284.68	0.489	-0.378	9	-2284.68	0.489	-0.378
1.00E-08	17	-2284.68	0.489	-0.378	12	-2284.68	0.489	-0.378
1.00E-10	19	-2284.68	0.489	-0.378	nc			
1.00E-12	19	-2284.68	0.489	-0.378	nc			
1.00E-14	19	-2284.68	0.489	-0.378	nc			
1.00E-16	21	-2284.68	0.489	-0.378	nc			

	Simplex				Genetic			
tolerance	conv.	α_2	ϕ_1	$\phi_{s,1}$	conv.	α_2	ϕ_1	$\phi_{s,1}$
1.00E-02	26	0.06	0.641	-0.364	116	-2279.07	0.488	-0.378
1.00E-04	60	0.06	0.638	-0.346	137	-2284.70	0.489	-0.378
1.00E-06	332	-2284.68	0.489	-0.378	170	-2284.68	0.489	-0.378
1.00E-08	354	-2284.68	0.489	-0.378	nc			
1.00E-10	385	-2284.68	0.489	-0.378	nc			
1.00E-12	395	-2284.68	0.489	-0.378	nc			
1.00E-14	405	-2284.68	0.489	-0.378	nc			
1.00E-16	nc				nc			

users of forecasting software consider ease of use, easily understandable results, and ease of interaction as the three most important features. These users do not realize, however, that the ultimate purpose of scientific software is to carry out calculations and return correct and reliable answers. As shown in this paper, there exist various sources of error in numerical computations, which can render computer generated results inaccurate and therefore invalid. Moreover, human errors or just plain ignorance by the software vendors can lead to inferior software that can produce seriously misleading results. Further, for implementing an econometric method such as ARIMA on a computer, there can be alternative approaches which can cause considerable discrepancies in the results across different programs and even within a single software system. Consequently, it is of great importance for researchers and practitioners to treat computer output with a degree of circumspection and verify computer generated forecasts manually or by using more than one software package where available.

We second McCullough (2000) that the software developers will supply accuracy only when users demand it. This is why, when choosing software, users should always demand from vendors proof of software accuracy and should never accept an unsubstantiated claim on this important matter. In general, it is the responsibility of the users to work only with vendors who genuinely care about software reliability through a demonstrated commitment to six best practices:

- *Document* online or in the user's manual the computational approach, algorithms used, and numerical limitations for each procedure.
- *Test* carefully the program before releasing it by using the standard benchmark datasets, well-known textbook models, and classic published results.
- *Accept* that no program is perfect and maintain an easily accessible "change-log" and a "bug-list" so that the users are aware of all modifications and known problems.
- *Respond* upon detection of new errors by issuing bug warnings and workarounds in order to prevent users getting answers that are wrong.
- *Correct* properly in the subsequent release all known accuracy flaws without attempting instead to make them less obvious.
- *Respect* all researchers' right to run the program and support research replication by offering an evaluation option taking into account that many software problems are discovered during a replication exercise.

Among the many different econometric and forecasting programs available today[10], some are are better than others in following the above mentioned principles for reliable scientific software. For example, Stata is known for its excellent documentation, which includes a three volume reference manual containing the algorithms for almost all procedures. The accuracy of TSP is tested thoroughly using a wide variety of benchmark models and datasets listed online at `http://www.stanford.edu/~clint/bench/`. SAS provides a periodic newsletter containing bug warnings and it releases on a timely basis hot fixes addressing various software issues. Moreover, as shown by studies such as Keeling and Pavur (2007) and McKenzie and Takaoka (2007), most software vendors do correct in the subsequent releases accuracy errors discovered by users. Finally, many if not all commercial programs have a time-limited trial version also.

Addinsoft offers a 30-day evaluation version of XLSTAT and we acknowledge the support team's efforts toward fixing the reported errors. The vendor also claims to test the program against other packages, however, we would like to know what package they tested the ARMA forecasts against because that program obviously has errors. Moreover, Addinsoft does not provide adequate documentation on computational details, and does not maintain for the program a change-log and a bug-list. Plus, considering how it took four attempts to correctly compute forecasts for an ARIMA model, and taking into account the software vendor's general tendency to conceal program flaws from the public, it is our understanding that Addinsoft's XLSTAT product currently fails to meet the requirements for reliable scientific software.

We would like to emphasize that, in this study, we tested the accuracy of XLSTAT's ARIMA estimation only. We do not know whether there exists additional flaws in the various other components of the program. Studies such as Knüsel (1998, 2002, 2005), McCullough and Wilson (1999, 2002, 2005), McCullough and Heiser (2008), and Yalta (2008) report

[10] See Renfro (2004) and Yurkiewicz (2003) for an overview of the existing econometric packages and forecasting packages respectively.

gross numerical errors in different versions of Microsoft Excel. Our study shows that "add-in" packages such as XLSTAT, which are designed to provide more and better functionality in Excel can also have errors of their own and thus need to be evaluated thoroughly with the existing benchmarks. Software testing is important as it reflects the ongoing concern of the user community regarding the reliability of commonly used programs providing statistical and econometric functionality.

Acknowledgments

We would like to express our appreciation to Thierry Famy from Addinsoft for responding kindly and quickly to our inquiries about XLSTAT. We also thank Allin Cottrell from Gretl, and Tom Doan from Estima for their comments and suggestions. The usual caveat applies.

References

Addinsoft. 2007. *XLSTAT Online Help*. URL `http://www.autobox.com/`.

Altman M, Gill J, McDonald MP. 2004. *Numerical Issues in Statistical Computing for the Social Scientist*. 1st edn., Wiley, New Jersey.

Automatic Forecasting Systems. 2007. *Autobox User's Guide*. URL `http://www.autobox.com/`.

Avriel M. 2003. *Nonlinear Programming: Analysis and Methods*. 1st edn., Dover Publishing, New York.

Bai J, Perron P. 2003. Computation and Analysis of Multiple Structural Change Models. *Journal of Applied Econometrics*, **18**: 1–22.

Berndt E, Hall B, Hall R, Hausman J. 1974. Estimation and Inference in Nonlinear Structural Models. *Annals of Social Measurement*, **3**: 653–665.

Box GEP, Jenkins GM. 1970. *Time Series Analysis, Forecasting, and Control*. 1st edn., Holden Day, San Francisco.

Cottrell A, Lucchetti R. 2008. *Gretl User's Guide*. URL `http://ricardo.ecn.wfu.edu/pub//gretl/manual/PDF/gretl-guide.pdf`, [Online; retrieved March 3, 2008].

Estima. 2007. *RATS Version 7 User's Guide*. URL `http://www.estima.com/`.

Keeling KB, Pavur RJ. 2007. A Comparative Study of the Reliability of Nine Statistical Software Packages. *Computational Statistics and Data Analysis*, **51**: 3811 – 3831.

Knüsel L. 1998. On the Accuracy of Statistical Distributions in Microsoft Excel 97. *Computational Statistics and Data Analysis*, **26**: 375–377.

Knüsel L. 2002. On the Reliability of Microsoft Excel XP for Statistical Purposes. *Computational Statistics and Data Analysis*, **39**: 109–110.

Knüsel L. 2005. On the Accuracy of Statistical Distributions in Microsoft Excel 2003. *Computational Statistics and Data Analysis*, **48**: 445–449.

Küsters U, McCullough BD, Bell M. 2006. Forecasting Software: Past, Present and Future. *International Journal of Forecasting*, **22**: 599–615.

Lovell MC, Selover DD. 1994. Econometric Software Accidents. *The Economic Journal*, **104**: 713–725.

Marquardt D. 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, **33**: 431–441.

McCullough BD. 1998a. Algorithm Choice for (Partial) Autocorrelation Functions. *Journal of Economic and Social Measurement*, **24**: 265–278.

McCullough BD. 1998b. Assessing the Reliability of Statistical Software: Part I. *American Statistician*, **52**: 358–366.

McCullough BD. 1999. Econometric Software Reliability: Eviews, LIMDEP, CHASM and TSP. *Journal of Applied Econometrics*, **14**: 191–202.

McCullough BD. 2000. Is It Safe to Assume That Software Is Accurate? *International Journal of Forecasting*, **16**: 349–357.

McCullough BD. 2004. Wilkinson's Tests and Econometric Software. *Journal of Economic and Social Measurement*, **29**: 261–270.

McCullough BD. 2008. Microsoft Excel's 'Not the Wichmann–Hill' Random Number Generators. *Computational Statistics and Data Analysis*, **52**: forthcoming.

McCullough BD, Heiser DA. 2008. On the Accuracy of Statistical Procedures in Microsoft Excel 2007. *Computational Statistics and Data Analysis*, **52**: forthcoming.

McCullough BD, Vinod HD. 1999. The numerical reliability of econometric software. *Journal of Economic Literature*, **37**: 633–655.

McCullough BD, Vinod HD. 2003. Verifying the Solution from a Nonlinear Solver: a Case Study. *The American Economic Review*, **93**: 873–892.

McCullough BD, Wilson B. 1999. On the Accuracy of Statistical Procedures in Microsoft EXCEL 97. *Computational Statistics and Data Analysis*, **31**: 27–37.

McCullough BD, Wilson B. 2002. On the Accuracy of Statistical Procedures in Microsoft Excel 2000 and Excel XP. *Computational Statistics and Data Analysis*, **40**: 713–721.

McCullough BD, Wilson B. 2005. On the Accuracy of Statistical Procedures in Microsoft Excel 2003. *Computational Statistics and Data Analysis*, **49**: 1244–1252.

McKenzie CR, Takaoka S. 2007. EViews 5.1. *Journal of Applied Econometrics*, **22**: 1145–1152.

Mélard G. 1984. Algorithm AS 197: A Fast Algorithm for the Exact Likelihood of Autoregressive-Moving Average Models. *Applied Statistics*, **33**: 104–114.

Newbold P, Agiakloglou C, Miller J. 1994. Adventures with ARIMA Software. *International Journal of Forecasting*, **10**: 573–581.

Renfro CG. 2004. A Compendium of Existing Econometric Software Packages. *Journal of Economic and Social Measurement*, **29**: 359–409.

Sanders NR, Manrodt KB. 2003. Forecasting Software in Practice: Use, Satisfaction, and Performance. *Interfaces*, **33**: 90–93.

Sawitzki G. 1994. Testing Numerical Reliability of Data Analysis Systems. *Computational Statistics and Data Analysis*, **18**: 269–286.

Stokes HH. 2004. On the Advantage of Using Two or More Econometric Software Systems to Solve the Same Problem. *Journal of Economic and Social Measurement*, **29**: 307–320.

Stokes HH. 2005. The Sensitivity of Econometric Results to Alternative Implementations of Least Squares. *Journal of Economic and Social Measurement*, **30**: 9–38.

US Census Bureau. 2007. *X-12-ARIMA Reference Manual*. URL `http://www.census.gov/ts/x12a/v03/x12adocV03.pdf`, [Online; retrieved March 3, 2008].

Wichmann BA, Hill ID. 1982. Algorithm AS 183: an Efficient and Portable Pseudo-Random Number Generator. *Applied Statistics*, **31**: 188–190.

Yalta AT. 2008. The Accuracy of Statistical Distributions in Microsoft ® Excel 2007. *Computational Statistics and Data Analysis*, **52**: forthcoming.

Yurkiewicz J. 2003. Forecasting Software Survey. *OR/MS Today*, **30**: 44–51.

Zeileis A, Kleiber C. 2005. Validating Multiple Structural Change Models – A Case Study. *Journal of Applied Econometrics*, **20**: 685–690.