A benchmark-free approach to access the accuracy of an OLS model calculation in an econometrically uncertain world

Houston H. Stokes
Department of Economics, University of Illinois, 601 South Morgan Street (M/C 144), Chicago, IL 60607, USA
Tel.: +1 773 643 4383; E-mail: hhstokes@uic.edu

Although it is possible to test the calculation accuracy of a software system's implementation of OLS using sample datasets that have been extensively studied and have benchmarked answers, this approach will not help access the calculation accuracy of a user's model for which such answers are not available. Since in theory estimated residuals of an OLS model are required to be orthogonal to the right-hand-side variables, this paper investigates the impact of changes in data precision and calculation method on the software's success in achieving this goal. In attempting estimation of an OLS model the investigator must make two critical decisions. The first decision is the precision of the data used in the calculation, both how it was loaded and whether its precision was subsequently increased for the actual calculation. The second and related decision is the calculation method. To investigate these tradeoffs, two rather simple and well known data sets that do not exhibit extreme multicollinearity are used to illustrate the data precision / calculation method accuracy tradeoff.

1. Introduction

In the uncertain world of econometric practice, care should be exercised in determining if OLS model calculations have been made using appropriate methods to achieve sufficient accuracy. This need for caution was famously driven home by Longley's [9] important paper in 1967 which implies serious accuracy problems can occur if estimation was attempted with mulicollinear datasets and sufficient precautions were not followed. The research methodology computational decision includes both deciding on the appropriate estimation method and selecting sufficient data precision to make the calculation possible.[1] As a basis for the argument to be presented here, it is assumed that in all cases the computer routines are top quality in their

[1] Altman-Micah-Gill-McDonald [1, p. 256] note "Surprisingly, inputting data into a statistical software package is not necessarily a straightforward process. Measurement errors occurs when the precision level of a statistical softweare package is exceeded. Observations may be silently truncated . . . " The above quote is concerned whether due to less that needed precision in the data loading step, the "wrong" problem is solved and/or the $X'X$ matrix rank is reduced. In the present paper we assume the problem implicit by the data loading precision and assess the accuracy of the resulting calculation.

0747-9662/08/$17.00 © 2008 – IOS Press and the authors. All rights reserved

implementation.[2] To demonstrate the quality of this code, absent a benchmark for the problem at hand, appropriate diagnostics should be performed to give confidence in the results. The suggested testing procedure which will be illustrated below, builds on the orthogonality assumptions implicit in OLS modeling and provides a test on the success of the calculation.

One of the key assumptions of the Classical Linear regression model, as discussed in Greene [13, pp. 10,24] among others since the 1960's classic texts by Goldberger [12] and Johnston [7], is the exogeneity of the independent variables, in the sense of these variables being distributed independently of the error term or else fixed in repeated sampling. In principle, under computationally ideal circumstances, the solution of the normal equations constructively imposes the restriction that the correlation between the residual error vector and each right hand side variable is constrained to be zero. However, in practice, this ideal mathematical result will not necessarily be obtained to a sufficient degree of accuracy. As will be demonstrated, if the data precision is not sufficient, the data matrix is not scaled, or the estimation method is not appropriate for the problem, the OLS estimated residuals can fail to be orthogonal to the equation's right-hand-side variables to an appropriate tolerance. This finding suggests the need to establish, as a diagnostic test, whether after a calculation orthogonality is sufficiently precisely achieved. Such a test can be especially useful in certain cases, such as when benchmark values for the estimated coefficients are unavailable, as indicated above.

More formally, it is well known that, in theory, if in the population the right hand side variables X of a linear model $y = X\beta + e$ are exogenous and X is full rank (the inverse of $X'X$ exists) ordinary least squares (OLS) can be an appropriate estimator of β. Furthermore, in this case OLS estimation of the sample coefficient vector $\hat{\beta} = (X'X)^{-1}X'y$ ideally minimizes the sum of squared sample errors $\hat{e}'\hat{e}$, in the process imposing the restriction that there is no correlation between the estimated residuals and the vectors of X, or $X'\hat{e} = 0$.[3] As shown for example by Greene [13, p. 20], if

[2] Subsequent to Longley's 1967 paper [9], in 1979 Dongarra, Moler, Bunch and Stewart [5] released the LINPACK subroutine library that became the gold standard for modern numerical calculation for many years. The importance of this release was that given the selected method, if the LINPACK matrix library was used the calculations would be performed as accurately as possible. While in 1961 a popular linear algebra book such as Hadley [6] did not mention the QR algorithm, in 1976 Strang's linear algebra book [22] did have these modern techniques covered and mentioned LINPACK explicitly since it had been circulating in test version form prior to its final release in 1979. In 1984 Longley [8] discussed Orthogonalization methods in detail to supplement material from his famous 1967 paper. In 1992 Anderson et al. [2] released LAPACK that further defined LINPACK, especially for large problems. Altman-Micah-Gill-McDonald [1] provides an up-to-date discussion of some of the technical issues from a modern perspective. Due to space limitations these will not be dealt with here.

[3] The inverse can be obtained with an LU factorization or marginally more accurately with a Cholesky factorization. The intrinsically more accurate QR approach proceeds by factoring $QR = X$ and is often used in less than ideal computational conditions. The condition number of a matrix $\lambda(X)$ is defined as the ratio of the largest to the smallest eigenvalue and provides insight into the problem. The larger the condition, the more difficult it is from a numerical standpoint to invert a matrix. Belsley et al. [3]

this result were obtained, the unexplained sum of squares could be reduced. However, the achievement of $X'\hat{e} \cong 0$ in practice is not necessarily assured: specifically, the numerical solution of the normal equations may be unsuccessful in imposing the restriction that the error term is orthogonal to the right hand side variables. [4] Thus at issue is the degree to which the actual calculations made are sensitive to the problem at hand, the data precision and the method of obtaining the OLS estimates.

Stokes [21] previously investigated a number of aspects of the OLS solution problem using analysis of highly collinear test data sets, in particular the Filippelli data set considered by Rogers et al. [18]. The focus of this research was to see how closely the estimated coefficients matched known test values for different estimation strategies. However, in the context of this research, the properties of the estimated residuals were not directly tested. In order to extend these results, the object in the present paper is to see how closely the estimated residual agrees with the theoretical orthogonal property of the residual against right-hand-side variables.

There are a number of good reasons to pursue this approach. First, test benchmarks are then not necessarily needed for the problem at hand, since given the machine precision, the more accurate the calculations the closer the estimated correlation between the estimated residual and the right-hand-side vectors will approach zero. Second the estimated correlation provides a way to generate a single number that can be used to compare the accuracy of the calculation of two different problems and in the process indicate if the data precision or method of estimation are appropriate for the problem at hand. In an OLS model containing k right-hand-side variables plus a constant, the maximum absolute correlation of the residual and one of the k right-hand-side variables indicates the worst case.

advises if the condition is $\geqslant 20$. There is cause for concern. It can be proved that $\lambda(X) = \sqrt{\lambda(X'X)}$. In situations when there is substantial multicollinearity in a regression model $\lambda(X)$ is already large. Forming $X'X$ just makes matters worse from a numerical sense and provides the rationali for use of the QR approach. After a *QR* factorization, Q is n by k and R is k by k upper triangular and, in fact, is a more accurate Cholesky factorization of $X'X$. Note that $R'R = X'X$. $QQ' \neq I$ and is n by n, while $Q'Q = I$ and is k by k. The QR approach thus gains accuracy over the more traditional approach by not having to explicitly form $X'X$. Using the QR factorization, $\hat{\beta} = R^{-1}Q'Y$. If the residual is all that is needed, then $\hat{e} = Y - QQ'Y$. This can be easily seen if we note that $\hat{e} = Y - X\beta = Y - QRR^{-1}Q'Y = Y - QQ'Y$. A less accurate, although mathematically equivalent, way to proceed is to calculate $\hat{e} = Y - X\beta = Y - XR^{-1}Q'Y$, which looses accuracy in the R^{-1} step, although the more accurate QR R can be used in place of the Cholesky R. In summary, the LU / Cholesky approach to OLS model estimation requires first forming the cross product matrix which inhenently causes accuracy losses that are avoided by the QR approach that proceeds by factoring X directly. The essential message, that will be illustrated with a number of example below, is that both computational techniques can make a substantial difference in the accuracy of an econometric calculation.

[4] If some of the right hand side variables are endogenous, then in the population they are related to the error term of the equation. Imposing the orthogonality restriction as is the case with OLS, thus causes a bias. Provided identification restrictions are met, in this case two stage least squares should be used in place of OLS. Two stage least squares replaces the right hand side endogenous variables with instrumental variables that it hopes are orthogonal to the error term. This issue is not the focus of the present paper.

2. Factors that impact estimation accuracy

There are number of factors that impact estimation accuracy, which will only be sketched at this point. These include the number of digits of precision with which numbers are represented in the data storage, how many significant digits on data input, and the particular method of calculation. The present paper is concerned mainly with investigating the effects of the precision with which the data are saved and the calculation method, although the importance of how many digits to report, given the input precision of the data, may also be of interest.

When a number is held in the computer, it is held with a certain degree of precision. At issue is the factors influence this decision. On conversion from single precision to double precision the computer will simply add some random digits during this process – as opposed to when real*8 is used every step of the way. At issue is how implicit assignment statements are handled. A Fortran program to investigate these ideas is shown in Table 1.[5] This program illustrates the fact that the code

```
real*8 x
x= 34.834702d-0
```

is not the same as

```
real*8 x
x= 34.834702e-0
```

Answers are given in Table 2 that were obtained after compilation using the Lahey/Fujitsu Fortran 95 Compiler Release 7.10.02. The data used in this example came from the NIST dataset distributed by the National Institute of Standards and Technology. This dataset is further discussed in Rogers et al. [18].

Five test problems were run for three assignment tests and three reading tests. The internal assignment tests for example 1 are discussed first.

- DIFF1, defined as `dlog(xd(i)) - dlog(dble(y(i)))`, is between a real*8 variable xd loaded from a statement such as `xd(1)= 34.834702d-0` and y which was loaded from an E number and was `2.486846548163719E-02`.
- DIFF2, defined as `dlog(x(i)) - dlog(dble(y(i)))`, replaces the x assignment with `x(1)= 34.834702e-0`. DIFF2 = `2.486845222146733E-02` which is quite different from DIFF1.
- DIFF3, defined as, `dlog(x(i)) - dlog(r8y(i)))`, compares x and r8y where r8y was read from a E number directly into a real*8 number. Since DIFF2 = DIFF3 it proves that the differences are not due to the use of the Fortran function dble.

[5] Although the Table 1 example is in Fortran, the argument could easily have been made using C or C++ code.

Table 1
Fortran Program to illustrate precision conversion issues

```
      real*8 x(3),xd(3),r8y(3),r8yd(3),diff1,diff2,diff3,diff4,diff5
      real*4 y(3)
      character*80 cwork1,cwork2,cwork3,cwork4

c     numbers taken from
c     Bennett, L., L. Swartzendruber, and H. Brown,
c                  NIST (1994).
c                  Superconductivity Magnetization Modeling.
c
c
c     cwork1     real*8 x
c     cwork2     real*4 x
c     cwork3     real*8 y
c     cwork4     real*4 y
c
      cwork1='34.834702d-0 34.393200d-0 34.152901d-0'
      cwork2='34.834702e-0 34.393200e-0 34.152901e-0'
      cwork3='33.979099d-0 33.845901d-0 33.732899d-0'
      cwork4='33.979099e-0 33.845901e-0 33.732899e-0'
c
      itestchar=1
c
      xd(1)=    34.834702d-0
      xd(2)=    34.393200d-0
      xd(3)=    34.152901d-0
      x(1)=     34.834702e-0
      x(2)=     34.393200e-0
      x(3)=     34.152901e-0

      y(1)=     33.979099E-0
      y(2)=     33.845901E-0
      y(3)=     33.732899E-0
      r8y(1)=   33.979099e-0
      r8y(2)=   33.845901e-0
      r8y(3)=   33.732899e-0
      r8yd(1)=  33.979099d-0
      r8yd(2)=  33.845901d-0
      r8yd(3)=  33.732899d-0

      write(6,*)'x    a real*8 variable loaded from a number E'
      write(6,*)'xd   a real*8 variable loaded from a number D'
      write(6,*)'y    a real*4 variable loaded from a number E'
      write(6,*)'r8y  a real*8 variable loaded from a number E'
      write(6,*)'r8yd a real*8 variable loaded from a number D'
      write(6,*)' '
      write(6,*)'Shows effect of setting a real*4 number in real*8'

      do i=1,3
      diff1=dlog(xd(i)) - dlog(dble(y(i)))
      diff2=dlog(x(i))  - dlog(dble(y(i)))
      diff3=dlog(x(i))  - dlog(r8y(i))
      diff4=dlog(x(i))  - dlog(r8yd(i))
      diff5=dlog(xd(i)) - dlog(r8yd(i))
```

Table 1, continued

```
      write(6,*)' '
      write(6,*)'Example  ',i
      write(6,*)'Test 1: xd - y    real 8-real*4 diff1',diff1
      write(6,*)'Test 2: x  - y    real 8-real*4 diff2',diff2
      write(6,*)'Test 3: x  - r8y  real*8-real*8 diff3',diff3
      write(6,*)'Test 4: x  - r8yd real*8-real*8 diff4',diff4
      write(6,*)'Test 5: xd - r8yd real*8-real*8 diff5',diff5
      enddo
c
      if(itestchar.ne.0)then
      write(6,*)' '
      write(6,*)'----------------------------------------------------'
      read(unit=cwork1,fmt=*)xd
      read(unit=cwork2,fmt=*)x
      read(unit=cwork3,fmt=*)r8yd
      read(unit=cwork4,fmt=*)y
      read(unit=cwork4,fmt=*)r8y
      write(6,*)'x    a real*8 variable loaded from a number E'
      write(6,*)'xd   a real*8 variable loaded from a number D'
      write(6,*)'y    a real*4 variable loaded from a number E'
      write(6,*)'r8y  a real*8 variable loaded from a number E'
      write(6,*)'r8yd a real*8 variable loaded from a number D'
      write(6,*)' '
      write(6,*)'* read does not differentiate between E and D'

      do i=1,3
      diff1=dlog(xd(i)) - dlog(dble(y(i)))
      diff2=dlog(x(i))  - dlog(dble(y(i)))
      diff3=dlog(x(i))  - dlog(r8y(i))
      diff4=dlog(x(i))  - dlog(r8yd(i))
      diff5=dlog(xd(i)) - dlog(r8yd(i))
      write(6,*)' '
      write(6,*)'Example  ',i
      write(6,*)'Test 1: xd - y    real 8-real*4 diff1',diff1
      write(6,*)'Test 2: x  - y    real 8-real*4 diff2',diff2
      write(6,*)'Test 3: x  - r8y  real*8-real*8 diff3',diff3
      write(6,*)'Test 4: x  - r8yd real*8-real*8 diff4',diff4
      write(6,*)'Test 5: xd - r8yd real*8-real*8 diff5',diff5
      enddo
      endif

      stop
      end
```

- DIFF4, defined as `dlog(x(i)) - dlog(dble(r8yd(i)))`, uses `x` and r8yd where r8yd loads from a D number and was `2.486847353604857E-02.`
- DIFF5, defined as defined as `dlog(xd(i)) - dlog(dble(r8yd(i)))`, was `2.486847353604857E-02` and should be the most accurate calculation since it is the difference between two real*8 numbers where both were read from D numbers.

The above examples shows how data loading effects the answers even with numbers well within ranges for real*4.[6] It suggests that software developers use caution in

[6] The above tests were run with the Lahey/Fujitsu Fortran 95 Compiler Release 7.10.02 on a Dell 650 workstation using Windows XP Professional. The same problem was run with the F77 and G77 compilers on a Linux workstation running RedHat with the same exact answers except that less digits were printed.

Table 2
Answers for assignment/read accuracy test

```
x    a real*8 variable loaded from a number E
xd   a real*8 variable loaded from a number D
y    a real*4 variable loaded from a number E
r8y  a real*8 variable loaded from a number E
r8yd a real*8 variable loaded from a number D

Shows effect of setting a real*4 number in real*8

Example   1
Test 1: xd - y    real 8-real*4 diff1 2.486846548163719E-02
Test 2: x  - y    real 8-real*4 diff2 2.486845222146733E-02
Test 3: x  - r8y  real*8-real*8 diff3 2.486845222146733E-02
Test 4: x  - r8yd real*8-real*8 diff4 2.486846027587871E-02
Test 5: xd - r8yd real*8-real*8 diff5 2.486847353604857E-02

Example   2
Test 1: xd - y    real 8-real*4 diff1 1.604095699681202E-02
Test 2: x  - y    real 8-real*4 diff2 1.604095468979484E-02
Test 3: x  - r8y  real*8-real*8 diff3 1.604095468979484E-02
Test 4: x  - r8yd real*8-real*8 diff4 1.604096914524724E-02
Test 5: xd - r8yd real*8-real*8 diff5 1.604097145226443E-02

Example   3
Test 1: xd - y    real 8-real*4 diff1 1.237394708902339E-02
Test 2: x  - y    real 8-real*4 diff2 1.237393818204299E-02
Test 3: x  - r8y  real*8-real*8 diff3 1.237393818204299E-02
Test 4: x  - r8yd real*8-real*8 diff4 1.237392964907340E-02
Test 5: xd - r8yd real*8-real*8 diff5 1.237393855605380E-02

------------------------------------------------------
x    a real*8 variable loaded from a number E
xd   a real*8 variable loaded from a number D
y    a real*4 variable loaded from a number E
r8y  a real*8 variable loaded from a number E
r8yd a real*8 variable loaded from a number D

* read does not differentiate between E and D

Example   1
Test 1: xd - y    real 8-real*4 diff1 2.486846548163719E-02
Test 2: x  - y    real 8-real*4 diff2 2.486846548163719E-02
Test 3: x  - r8y  real*8-real*8 diff3 2.486847353604857E-02
Test 4: x  - r8yd real*8-real*8 diff4 2.486847353604857E-02
Test 5: xd - r8yd real*8-real*8 diff5 2.486847353604857E-02

Example   2
Test 1: xd - y    real 8-real*4 diff1 1.604095699681202E-02
Test 2: x  - y    real 8-real*4 diff2 1.604095699681202E-02
Test 3: x  - r8y  real*8-real*8 diff3 1.604097145226443E-02
Test 4: x  - r8yd real*8-real*8 diff4 1.604097145226443E-02
Test 5: xd - r8yd real*8-real*8 diff5 1.604097145226443E-02

Example   3
Test 1: xd - y    real 8-real*4 diff1 1.237394708902339E-02
Test 2: x  - y    real 8-real*4 diff2 1.237394708902339E-02
Test 3: x  - r8y  real*8-real*8 diff3 1.237393855605380E-02
Test 4: x  - r8yd real*8-real*8 diff4 1.237393855605380E-02
Test 5: xd - r8yd real*8-real*8 diff5 1.237393855605380E-02
```

internal assignment statements that, except for open source code, are hidden from the user. If data is to be initialized into a real*8 variable, then D not E format should be used.

The next set of examples uses the Fortran internal * read to test if an E number read into a real*8 is seen by the run time I/O routines as real*8. The fact that DIFF1=DIFF2 and DIFF3=DIFF4=DIFF5 offers convincing evidence that data using E format can be read into real*8 variables without an accuracy loss due to conversion.

The above examples document the accuracy loss when calculations are made using real*8 data that is first read into a real*4 variable. Such data will have measurement error that was not detectable by any testing of the correlation between the residual and the right had side vectors since the OLS solution world attempt to force the residual to be orthogonal to the data that contains the measurement error.

It should be noted that a distinction has to be made between real world data and data sets that are created in order to be used for benchmark tests. In the case of real world data, we know that rounding has occurred, and that the data as obtained are generally rather imprecise. In contrast, in the case of created (test) data, we assume the precision of the input data. In the NIST data sets of Rogers et al. [18], the implicit assumption is that all m digits reported in the input data are accurately known and that trailing data points are exactly 0. This can only be assumed true for the problem input series. In a model such as Filippelli $y = f(x, x^2, \ldots, x^{10})$ where data has to be built before the calculation, only y and x are assumed to be explicitly known to m digits.[7] The accuracy of the other variables in the model are determined by the precision of the data calculation forming (x^i for $i > 1$). Hence in this case we have exact data input for only y and x with calculation errors induced in x^i for $i > 1$. Since the exact data is input into a floating point representation, the size of the mantissa (real*8 or real*4) in the floating point examples discussed later impacts the accuracy of subsequent calculations.[8] An alternative to floating point calculation is to use variable precision mathematics where (x^i for $i > 1$) can be calculated to any degree of accuracy. For further information on these issues see Altman-Gill-McDonald [1, chapter 2].

Whether this example generalizes to all compilers is an open question but one that should be addressed by software developers.

[7] For the Filippelli data y is given to 4 digits and x to 10 digits. Stokes [21] investigates this problem in some detail using alternative estimation methods and alternative data precision.

[8] Altman-Gill-McDonald [1, pp. 23–29] have a comprehensive discussion of some of the problems of computer arithmetic. Using their notation, floating point numbers are represented as $y = \pm m \times \beta^{e-1}$ where m is the mantissa and β is the base. A IEEE single precision number (REAL*4 in Fortran) has a relatively limited size mantissa that permits only 5–7 digits of representation. In contract, a double floating point number (REAL*8 in Fortran) has a larger mantissa that permits 12–14 digits of representation. Since most numbers used in economic analysis do not get outside the real*4 range, it is the size of the mantissa that causes number accuracy issues.

Table 3
Vector (y) and matrix (X) norms and matrix (X) condition estimates

Norm/Condition	Vector	Matrix
One-Norm	$\|y\|_1 = \Sigma_i \|y_i\|$	$\|X\|_1 = \max_j \Sigma_i \|x_{ij}\|$
p-Norm	$\|y\|_2 = (\Sigma_i \|y_i\|^2)^{.5}$	$\|X\|_2 = \max_i s_i$ where s_i singular value of X
Infinity-Norm	$\|y\|_\infty = \max_i \|y_i\|$	$\|X\|_\infty = \max_i \Sigma_j \|x_{ij}\|$
Frobenius norm	$\|y\|_F = \|y\|_2$	$\|X\|_F = \sqrt{(\Sigma_{ij} \|x_{ij}\|^2)} = \sqrt{\Sigma_{ii}(diag(x'x))}$
Condition		$\|X\|_p \|X^{-1}\|_p \, p = 1, 2, \infty, F$

3. Numerical complexity of a problem

The prior section has focused on the precision of the number storage as a limiting factor in calculation accuracy. Another important factor impacting the accuracy of the final calculation is whether the number to be saved can be exactly represented as a factor of 2. The number 0.1, which has an infinitely repeating binary representation, is a example of a number that cannot be exactly represented. In addition to these representation issues is the numerical complexity of the data to be analyzed. There are a number of measures that address this issue. A good reference is the LAPACK guide, Anderson, Bai, Biscof et al. [2] and Moler [11], that illustrate the use of vector and matrix norms to classify the numerical complexity of a problem and the expected calculation accuracy. The suggested measures are listed in Table 3.

Moler [11, pp. 67–72] shows that given a system $X\beta = y$, following Wilkinson's pioneering analysis, the relative residual is $\frac{\|u - X\hat{\beta}_*\|}{\|X\|\|\hat{\beta}\|} \leqslant p\varepsilon$ and the relative error in the coefficient is $\frac{\|\hat{\beta} - \hat{\beta}_x\|}{\|\hat{\beta}_x\|} \leqslant p\kappa(X)\varepsilon$, where ε is the relative machine precision that is a function of whether real*8 or real*4 data storage is being used, $\kappa(X)$ is the matrix condition and p is usually set as 10. $\|X\|$ can be calculated with any of the four methods listed in Table 1, although $\|X\|_2$ is most common. Define $\hat{\beta}$ as the true/exact coefficient vector while $\hat{\beta}_*$ is the calculated coefficient vector.[9] The importance of this discussion is that it shows how accuracy bounds of the error term and the estimated coefficient are a function of both the characteristic of the problem, as measured by the condition number of and the precision of the data storage as measured by . For real*8 the Fortran epsilon function returns 2.220446049250313E-16 while for real*4 and real*16 the corresponding values are 0.19209290E-07 and 1.9259299443872358530559779425849273E-0034, respectively. [10] Given the data precision, absolute values of the correlation of the residual and a right-hand-side variable less than or equal to these values should be assumed to be 0.0. Such

[9] The term "true" does not mean population coefficient. What is meant is the calculated parameter if variable precision math were used to obtain an exact answer given the data.

[10] This is often confusion between what is returned by the Fortran function epsilon( ) in contrast to the Fortran function tiny( ). The first being a positive value that is almost negligible compared to unity, while the second is the smallest representable positive number of data type. For example the Fortran program

residuals have met the orthogonality assumption of OLS.[11] Assume X is an n by k matrix of right-hand-side variables whose condition is $\kappa(X)$. The condition number of $X'X$ can be shown to be $[\kappa(X)]^2$, which makes a difficult problem harder and is a major motivating reason for using the QR approach to estimate an OLS model rather than explicitly forming $X'X$. If $X'X$ is needed, R from the QR should be used in place of the Cholesky factorization of $X'X$. Detailed discussion of these issues will not be pursued further, since the major motivation of this paper is to illustrate by example the magnitude of the accuracy problem if the data storage precision is not sufficient. By design, what appear to be relatively simple problems are used so that the reader will see that the issues raised in the paper will be applicable to researchers solving everyday problems.

4. Testing residual orthogonality with the right-hand-side variables

A comparison between the accuracy of single precision (real*4) and double precision (real*8) calculation in imposing the restriction $E(e, x_i) = 0$ will be illustrated using two representative data sets. One is the GE equation of the famous Grunfeld [16] data, discussed by Theil [24] and recently used as the basis for a number of benchmarks of various regression diagnostic tests by Renfro [17], which involves two variables on the right plus the constant. This problem appears to have minimal

```
 real*4 r4
 real*8 r8
 real*16 r16
 write(6,*)'Real*4    epsilon   ',epsilon(r4)
 write(6,*)'Real*8    epsilon   ',epsilon(r8)
 write(6,*)'Real*16   epsilon   ',epsilon(r16)
 write(6,*)'Real*4    tiny      ',tiny(r4)
 write(6,*)'Real*8    tiny      ',tiny(r8)
 write(6,*)'Real*16   tiny      ',tiny(r16)
 stop
 end
```

produces

```
 Real*4   epsilon   1.19209290E-07
 Real*8   epsilon   2.220446049250313E-16
 Real*16  epsilon   1.925929944387235853055977942584927 3E-0034
 Real*4   tiny      1.17549435E-38
 Real*8   tiny      2.225073858507201E-308
 Real*16  tiny      3.362103143112093506262677817321752 6E-4932
```

[11] Fortran code for testing if x is in fact 0.0 for real*8 should be performed as

```
if((x+1.0d+00).eq.1.0d+00)write(6,*)'x is in fact 0.0'
```

The logically equivalent command

```
if(x.eq.0.0d+00)write(6,*)'x is in fact 0.0'
```

will not work properly. The reason is that when x is a very small number when it is added to 1.0d+00 the result is seen as 1.0d+00. The direct comparison to 0.0d+00 will not detect that the number x is too small to be meaningful. For further detail see Dongarra et. al. [5] which recommends this calculation.

difficulty and is the best case for data saved in real*4. The second problem involves estimating a 3 lag model of the gas data studied by Box and Jenkins [4] and later by Tiao and Box [23]. While their analysis involved 6 lags, 3 lags were selected to illustrate problems that occur, even in a reduced and thus presumably easier model. [12] The object is to investigate the relationship between the precision of the data (real*4, real*8, real*16 and variable precision arithmetic [VPA]), the method of solution (Cholesky/LU vs QR) and the difficulty of the problem. More detail on this issue is contained in Stokes [19–21], Altman, Gill and McDonald [1] and an important paper by McCullough and Vinod [10].

The first problem based on the Grunfeld [14] data is relatively simple in that it does not involve right-hand-side variables of widely different sizes. The mean of General Electric (GE) value of the firm (GEF) was 1941.33 (SD = 413.84), while the mean of GE stock of plant and equipment (GEC) was 400.160 (SD = 250.619). The second example uses a three-lag model of the gas data that involved different size variables on the right. The mean for GASIN was -0.0568345 (SD = 1.0728), while the mean for GASOUT was 53.5091 (SD = 3.202). A summary of the findings is given in Table 4.

Using the Grunfeld [14] data set and real*8 data, the correlations for the Cholesky estimation were [-0.356987E-14 -0.260707E-14] vs slightly more accurate results of [0.271906E-15 0.333125E-15] for the QR approach. The corresponding numbers for real*16 are [-0.852072E-33 -0.188439E-33] and [-0.912934E-34 -0.263815E-33], respectively. Using the B34S variable precision arithmetic (VPA) default settings, the correlations were [-0.332843M-62 -0.179417M-62] for a model estimated using the LU method to invert where M is the VPA exponent that is similar to E. The number of digits for the VPA calculation was set to the default 64 for illustration purposes. Using real*4 data storage these correlations increased to [0.171420E-05 0.126598E-06] and [0.216786E-06 -0.236300E-06] for LU and QR estimated models respectively.[13] The importance of this example is that it shows that even when using a very widely known and simple data set, the impact of data precision on calculation accuracy is marked. These values are roughly 10 times the epsilon values for the data precision of the data and are to be expected. The coefficient vectors displayed as rows for real*8, real*16 and real*4 were, respectively,

```
real*8  Chol 0.265512E-01 0.151694 -9.95631
real*16 Chol 0.265512E-01 0.151694 -9.95631
real*4  LU   0.265513E-01 0.151694 -9.95650
```

[12] Tiao and Box [23] used this data set to illustrate VAR and VARMA models. The unconstrained VAR identification model contained 6 lags or 13 coefficients per row.

[13] The accuracy of the LU and Cholesky methods are similar and less than what can be obtained with the QR approach. The lines Real*8 LU and Real*8 Chol show what occurs using these methods and should be compared to Real*8 QR. The reason for this test is that the Real*4 tests were done with the LINPACK LU routines not Cholesky routines.

Table 4

Correlation between the error term and right-hand-sides

```
Grunfeld Example

               GEF              GEC
_____________________________________________________________________________________
Real*8  Chol -0.356987E-14  -0.260707E-14
Real*8  LU   -0.912202E-15  -0.796604E-16
Real*8  QR    0.271906E-15   0.333125E-15
Real*16 Chol -0.852072E-33  -0.188439E-33
Real*16 QR   -0.912934E-34  -0.263815E-33
Real*4  LU    0.171420E-05   0.126598E-06
Real*4  QR    0.216786E-06  -0.236300E-06
VPA           -.332843M-62   -.179417M-62

________________________________________________________________________________________________________________

Model Estimated:  GE Investment  = -9.9563  + .026551 Value GE   + .15169 GE Stock plant & Equipment
                                   (-.317)   (1.706)               (5.9015)
                                                                             e'e                  =132165877
                                                                             1/Cond. X'X          = 8.36336e-9
________________________________________________________________________________________________________________
Gas Data of lag Order 3
________________________________________________________________________________________________________________
Real*8  Chol -0.119284E-10   0.868642E-11  -0.430643E-11  -0.434085E-13   0.455703E-13  -0.123613E-13
Real*8  LU    0.949846E-10   0.803432E-10   0.903182E-10  -0.367528E-10  -0.454223E-10  -0.561377E-10
Real*8  QR    0.131642E-13   0.120757E-13   0.106601E-13  -0.596223E-14  -0.858368E-14  -0.108972E-13
Real*16 Chol -0.132839E-28   0.747619E-29  -0.214679E-29   0.515527E-32   0.797312E-32   0.441218E-32
Real*16 QR   -0.137433E-31  -0.126724E-31  -0.112671E-31   0.584987E-32   0.889735E-32   0.117243E-31
Real*4  LU   -0.895673E-01  -0.918947E-01  -0.891362E-01   0.369165E-01   0.452364E-01   0.557397E-01
Real*4  QR   -0.144533E-05  -0.151427E-05  -0.153826E-05   0.124983E-06   0.420500E-06   0.724758E-06
VPA            .957024M-59    .796435M-59    .628011M-59   -.691398M-59   -.839960M-59   -.963458M-59

__________________________________________________________________________________________________________________

Model Estimated: gasout = f(gasout(t-1),…,gasout(t-3),gasin(t-1),……,gasin(t-3), constant)

      1.61227      -0.934274       0.194382       0.105623      -0.321536      -0.194984        6.80967
      (28.147)     (-10.70)        (4.65)         (1.36)        (-2.11)        (-1.83)          (12.02)

                                                                             e'e                  = 18.0588
                                                                             1/Cond.X'X           = 7.928195e-8
```

For data descriptions, see text. All calculations are made with the B34S program version 8.11D. The Cholesky calculation was made with various precisions of the LINPACK Cholesky routines. The QR results were run with the LINPACK QR code, except for the real*4 function call to the qr( ) command, which used LAPACK. The VPA results were done with the LINPACK routines that were converted to run VPA math. The real*8 LU and real*4 LU calculations were made with LINPACK routines DGECO DGEDI and SGECO SGEDI, respectively.

with relatively minor differences showing up for the real*4.

For the somewhat more complex gas data set, a different pattern emerges in the bottom of Table 4. For real*8 the correlations increase for all cases, with the QR estimated models showing substantially less correlation. All correlations, however, are substantially above the machine epsilon values for the data storage precision used. What is surprising is the relatively poor real*4 showing of correlations in the area of $|$xE-01$|$ for models solved using the LU factorization. Use of the QR method of OLS calculation reduces these correlations to the range $|$xE-05$| - |$xE-06$|$, which is still substantially above epsilon for real*4 i.e. $\sim$0.192E-07. The importance of this finding is that it illustrates that any calculation involving real*4 data is very prone to error and should be avoided. However if real*4 data are used, it is highly recommended that the QR method be used. Even with the real*16 data, gains from using the QR are obtained.

Table 5 replicates the calculations in Table 4 for real*8 and real*4 data using Matlab 2007a, while Table 6 attempts the same problem using Matlab 2006b. Note that there are very minor differences indicating that in these two releases of Matlab there appear to be hidden and not announced changes that marginally impacted accuracy. [14]

[14] The Matlab 2006b results were done on a Windows XP Professional Dell 650 work station with dual Xeon^{TM} 3.0 GH processors. The Matlab 2006b results were done on a Dell Latitude Windows XP

Table 5
Correlation of the residual and right-hand-side variables using matlab 2007a

Grunfeld Example

	GEF	GEC
Real*8 LU	-1.2559e-015	4.3021e-016
Real*8 QR	8.3267e-017	2.0817e-017
Real*4 LU	-4.4651e-007	1.4417e-007
Real*4 QR	-9.9298e-008	2.4814e-007
Real*4 QR_2	-4.7489e-009	1.0605e-007

Gas Data of lag Order 3

Model Estimated: gasout = f(gasout(t-1),…,gasout(t-3),gasin(t-1),……,gasin(t-3), constant)

Real*8 LU	2.4454e-010	2.4313e-010	2.233e-010	-1.1906e-010	-1.4591e-010	-1.7591e-010
Real*8 QR	-1.4745e-015	-1.0686e-015	-1.027e-015	-8.9304e-015	-4.5935e-015	2.9837e-016
Real*4 LU	-0.022544	-0.023669	-0.026415	0.0035028	0.0054714	0.0087814
Real*4 QR	2.6992e-006	4.2817e-006	5.239e-006	2.8188e-006	1.8091e-006	9.9824e-007
Real*4 QR_2	8.7203e-005	8.8714e-005	8.4771e-005	-2.5406e-005	-3.4171e-005	-4.6354e-005

All calculations done with Matlab version 2007a. Correlation is done with the Matlab supplied command corr. The LU factorization and QR analysis is done in Matlab with the LAPACK software [2]. Real*4 QR calculates the residual as $e = y - Q'Qy$, while Real*4 QR_2 uses $e = y - x\beta = y - xR^{-1}Q'y$. Real*8 data were converted to real*4 in Matlab using the built-in function single( ).

Table 6
Correlation of the residual and right-hand-side variables using matlab 2006b

Grunfeld Example

	GEF	GEC
Real*8 LU	2.0123e-016	-4.4409e-016
Real*8 QR	4.9613e-016	2.6368e-016
Real*4 LU	-1.8515e-006	-3.2783e-007
Real*4 QR	-2.9802e-008	2.9802e-007
Real*4 QR_2	-7.0781e-008	4.4703e-008

Gas Data of lag Order 3

Model Estimated: gasout = f(gasout(t-1),…,gasout(t-3),gasin(t-1),……,gasin(t-3), constant)

Real*8 LU	1.257e-010	1.3066e-010	1.2848e-010	-5.5661e-011	-6.6412e-011	-7.9434e-011
Real*8 QR	-1.2436e-014	-1.1942e-014	-1.2615e-014	2.2346e-014	2.2511e-014	2.1032e-014
Real*4 LU	-0.033102	-0.033387	-0.032592	0.015702	0.019042	0.022815
Real*4 QR	2.0284e-006	1.682e-006	1.3784e-006	-2.6099e-006	-2.5911e-006	-2.6249e-006
Real*4 QR_2	-1.2402e-005	-1.1405e-005	-1.027e-005	1.1327e-005	1.2809e-005	1.3571e-005

For a discussion of what is calculated, see Table 5. Matlab documentation indicates that the LAPACK routines used were DLANGE, DGETRF, DGECON, DGETRI and SLANGE, SGETRF, SGECON, SGETRI for real*8 and real*4, respectively for the inv( ) command and DGEQRF, DORGQR and SGEQRF, SORGQR for qr( ).

However, the patterns exhibited in both tables are similar and support the results obtained with B34S reported in Table 2. Using Matlab 2007a for real*4 gas data and the LU method of estimation produced correlations between $|-0.0264|$ and 0.0035, which are way larger than epsilon for this data type. Since most regression packages use LU or Cholesky to solve OLS, this finding shows the danger, since the calculation of the model has broken down. Both versions of Matlab produced error messages to warn the user when the data was real*4. For Matlab 20006b the error message was "Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND $=$ 5.224432e-008", which should alert users to problems. Matlab is to be commended for providing the user input on rank problems of estimation. Many software systems do not provide this warning. For the GE problem the message was "Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND $=$ 8.358012e-009." The exact Matlab commands used for the gas data problem are listed below. The command getb34s brings in the x and y matrix with the appropriate lags for the two problems studied.

Professional machine.

```
x=getb34s(’xdata2.m’);
y=getb34s(’ydata2.m’);
beta8=inv(x’*x)*x’*y;
res8=y-x*beta8;
yhat_8=x*beta8;
newx8=x;
newx8(:,7)=res8;
disp(’Matlab LU real*8 results’)
c8=corr(newx8)

beta4=inv(single(x)’*single(x))*single(x)’*single(y);
disp(’++++++++++++++++ beta4-beta8 +++++++++++++++++++’)
beta4-beta8

yhat_4=single(x)*beta4;
res4 =single(y)-single(x)*beta4;
newx4=single(x);
newx4(:,7)=res4;
disp(’Matlab LU real*4 results’)
c4=corr(newx4)

betatest=[beta8 double(beta4) ]

% qr
disp(’Matlab QR real*8 and QR real*4’)
[q8,r8]=qr(x,0);
[q4,r4]=qr(single(x),0);
yhat_q8=q8*q8’*y;
yhat_q4=q4*q4’*single(y);
% plot(yhat_q8-yhat_q4)
% [yhat_q4 yhat_q8 yhat_8 yhat_4]
res_q8= y -yhat_q8;
res_q4=single(y)-yhat_q4;

% Get residual another way via beta to test
beta_q4=inv(r4)*q4’*single(y);
% ’
yhat_q4_alt=single(x)*beta_q4;
res_q4_alt=single(y)-yhat_q4_alt;
disp(’beta_q4-beta8’)
beta_q4-beta8

newx8_q=x;
```

```
newx4_q=single(x);
newx8_q(:,7)=res_q8;
newx4_q(:,7)=res_q4;
c8_q=corr(newx8_q);
c4_q=corr(newx4_q);

disp('alt c4_q via beta')
newx4_q_alt=single(x);
newx4_q_alt(:,7)=res_q4_alt;
c4_q_alt=corr(newx4_q_alt);

testres=[res8 res_q8 res4 res_q4 res_q4_alt];
testaccr =[c8(7,:)' c8_q(7,:)' c4(7,:)' c4_q(7,:)'
c4_q_alt(7,:)']';
testaccr
disp('++++++++++++++++ Matlab Ending +++++++++++++++')
quit
```

Table 7 lists various summary measures to describe the problems studied using the measures discussed in Table 3. These calculations were made with B34S and validated in Matlab. Note that the condition 2 values for x for the Grunfeld and gas data were 10,179.3794 and 3666.0302, respectively, and is the square root of the cond 2 values for $X'X$ of 103619764.9 and 13439777.78 respectively. Table 3 defines condition 2 in terms of the $\|X\|_2$, which is the maximum singular value of X.[15] What is a bit surprising is that the condition number of both X and $X'X$ for the Grunfeld data are larger than what was found for the gas data. Accuracy of meeting the residual orthogonality condition cannot always be determined from the condition number of the two matrices. These findings suggest that the residual correlation test provides a needed additional diagnostic statistic, that it is imperative to calculate if real*4 data are used and highly desirable otherwise.

5. Conclusion

These examples discussed in this paper document the dangers of estimating parameters using real*4 data and real*4 calculation. For OLS models, the probability of inaccuracy is especially great when the LU estimation method is used as opposed to the QR estimation approach. The best advice is to have both the data and the calculations 100% in real*8 and use the QR method if there is any indication of

[15] Note that singular values can be calculated on a rectangular matrix. Given that s_i, $i = 1, \ldots, k$ are the singular values of X ordered from largest to smallest, $\kappa(X)_2$ can be calculated as s_1/s_k.

Table 7
Summary statistics for grunfeld and gas data problems

Test	Grunfeld	Gas data	Ratio
Norm 1 xpx	94437564.78	2543686.426	0.026935
Norm 2 xpx	81840116.56	2523291.716	0.030832
Norm 3 xpx	94437564.78	2543686.426	0.026935
Norm 4 xpx	81848691.77	2523291.941	0.030829
Cond 1 xpx	119645470.2	19085694.38	0.159519
Cond 2 xpx	103619764.9	13439777.78	0.129703
Cond 3 xpx	119645470.2	19085694.38	0.159519
Cond 4 xpx	103630622.2	13493508.94	0.130208
Cond LAPACK	119645470.2	19085694.38	0.159519
Cond LINPACK	119569773.2	12613211.34	0.105488
Eigen max/min	103619767	13439777.75	0.129703
Norm 1 x	38826.5	15674.3	0.403701
Norm 2 x	9046.552745	1588.487241	0.17559
Norm 3 x	3649.8	185.953	0.050949
Norm 4 x	9111.799002	1588.904334	0.174379
Cond 2 x	10179.3794	3666.030248	0.360143

$\|X\|_i =$ *Norm i*; $\|X\|_i\|X^{-1}\|_i =$ *Cond i*.
For further discussion see Table 3.
Ratio = Gas Data / Grunfeld.

rank problems in the data set. While the exposition of this paper has stressed OLS estimation, the tests proposed here can be applied to test the calculation accuracy of models estimated with the second stage of two stage least squares since in that case the instrumental variables on the right hand side should be orthogonal to the error term. Many other estimation methods also require solution of the normal equations that require the errors to be orthogonal to the X matrix columns and can benefit from use of the proposed benchmark-free calculation test The evidence presented suggests that the condition number of the X matrix should be checked, but that this diagnostic test is not a 100% reliable indicator of potential problems. Inspection of whether the estimated residual is orthogonal to the right-hand-side variables is a simple test that does not require benchmarks of the model and can be employed as an early warning system of potential accuracy problems in estimating the coefficients of a variety of models. It goes without saying that the correlation routines used to perform the proposed test must be highly accurate and not exhibit the problems documented by McCullough and Vinod [10] in Table 2 of their classic paper.

Acknowledgements

Diana A. Stokes provided needed editorial assistance. Suggestions from Bruce McCullough and especially from Charles Renfro have been most helpful. Arnold Zellner raised a number of important issues for future research regarding the impact of the number of digits in the original data on any statistical inference independent of any calculation or data precision considerations. This topic, while somewhat outside

the main thrust of this paper, was noted as an important topic for future research. The author is responsible for any remaining errors.

References

[1] M. Altman, J. Gill and M. McDonald, *Numerical Issues in Statistical Computing for the Social Scientist*, Wiley, New York, 2004.

[2] E. Anderson, Z. Bai, C. Bishof, J. Demmel, J. Dongarra, J. DCu Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorenson, *LAPACK User's Guide*, SIAM, Philadelphia, 1992.

[3] D. Belsley, E. Kuh and R. Welsh, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity,* New York: John Wiley and Sons, 1980.

[4] G.E.P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control,* rev. ed., San Francisco: Holden Day, 1976.

[5] J. Dongarra, C.B. Moler, J.R. Bunch and G.W. Stewart, *LINPACK User's Guide*, SIAM, Philadelphia, 1979.

[6] G. Hadley, *Linear Algebra,* Reading, Massachusetts: Addison-Wesley Publishing Company, 1961.

[7] J. Johnston, *Econometric Methods,* New York: McGraw-Hill, 1963, 1972, 1984.

[8] J. Longley, *Least Squares Computations Using Orthogonalization Methods,* New York: Marcel Decker, 1984.

[9] J. Longley, An Appraisal of Least Squares Programs for the Electronic Computer From the Point of View of the User, *Journal of the American Statistical Association* **62**(319) (1967), 819–841.

[10] B.D. McCullough and H.D. Vinod, The numerical reliability of econometric software, *Journal of Economic Literature* **37** (June 1999), 633–665.

[11] C.B. Moler, *Numerical Computing with Matlab*, SIAM, Philadelphia, 2004.

[12] A.S. Goldberger, *Econometric Theory*, New York: Wiley, 1964.

[13] W.H. Greene, *Econometric Analysis,* 5th ed., New Jersey, Prentice Hall, 2003.

[14] Y. Grunfeld, *The determinants of corporate investment,* unpublished Ph.D thesis, The University of Chicago, 1958.

[15] C. Renfro, Econometric software: The first fifty years as perspective, *Journal of Economic and Social Measurement* **29**(1–3) (2004), 9–107.

[16] C. Renfro, Normative considerations in the development of a software package for econometric estimation, *Journal of Economic and Social Measurement* **23** (1997), 277–330.

[17] C.G. Renfro, *The Practice of Econometric Theory,* Heidelberg: Springer, 2009.

[18] J. Rogers, J. Filliben, L. Gill, W. Guthrie, E. Lagergren and M. Vangel, *StRD: Statistical Reference Statistical Reference Data Sets for Assessing the Numerical Accuracy of Statistical Software,* NIST TN # 1396, National Institute of Standards and Technology, 1998.

[19] H.H. Stokes, *Specifying and Diagnostically Testing Econometric Models,* (2nd ed.), Quorum Press, Westport, Conn, 1997.

[20] H.H. Stokes, On the advantage of using two or more econometric software systems to solve the same problem, *Journal of Economic and Social Measurement* **29**(1–3) (2004), 307–320.

[21] H.H. Stokes, The sensitivity of econometric results to alternative implementations of least squares, *Journal of Economic and Social Measurement* **30** (2005), 9–38.

[22] G. Strang, *Linear Algebra and Its Applications,* New York: Academic Press, 1976.

[23] G. Tiao and G.E.P. Box, Modeling Multiple Time Series with Applications, *Journal of American Statistical Association* **76** (1981), 802–816.

[24] H. Theil, *Principles of Econometrics,* Wiley, New York, 1971.