

Experience with the StRD: Application and Interpretation

B. D. McCullough
Federal Communications Commission
Washington, D. C. 20554
bmcullo@fcc.gov

Abstract

NIST produced the StRD using multiple precision, but PC users generally are limited to double precision. What type of accuracy PC users can expect for linear problems is examined numerically. An excursus on the QR/SVD observes that the former usually provides more accurate digits. Desired performance on nonlinear problems is discussed. Additionally, some finer points of applying the StRD are noted, such as the existence of multiple solutions for linear and nonlinear problems.

1 Introduction

The StRD has been applied to statistical (McCullough, 1998, 1999), econometric (McCullough, 1999a; Vinod and Silverio, 1999) and spreadsheet software (McCullough and Wilson, 1999), each time uncovering numerous flaws. NIST produced the StRD (available at <http://www.nist.gov/itl/div898/strd>) using multiple precision and produced its “certified values” by rounding the solutions to 15 digits (for linear problems) or 11 digits (for nonlinear problems). Complete details can be found in Rogers, *et al* (1998). PC users generally are limited to double precision, and so the question naturally arises, “What type of accuracy can a PC user expect?” In some cases, 8 digits of accuracy indicates that the software performs very well, and other times eight digits of accuracy can indicate exceedingly poor software. This question depends in part on the particular brand of software and the algorithms it implements, but also on the operating system. For example, one popular package returns two different answers to $\text{INT}(\text{SQRT}(25))$ depending upon whether it is run on UNIX/LINUX (5) or Microsoft Windows (4). The difference is not due to the software, but the way the operating system handles basic mathematical operations.

Determining the level of accuracy a PC user can expect is further complicated by the fact that companies customarily do not reveal source code, and may poorly

implement an algorithm. Upon occasion, a package may claim to use one algorithm when, unbeknownst to the user, an inferior algorithm actually is implemented. In some sense, interpreting benchmark results is akin to the classic “black box” problem faced by engineering students. Nonetheless, useful insight can be gained. In the cases of the univariate and linear regression suites, existing algorithms can be implemented in Fortran, and these results can be compared to the StRD certified values. In the cases of the ANOVA and nonlinear least squares suites, the variety of algorithms available to solve the problems is too great to permit the above approach. However, some numerical results for ANOVA are instructive, and a qualitatively-oriented discussion of nonlinear is fruitful. Each of the four suites is discussed, in turn. Additionally, some finer details of applying the StRD are mentioned.

2 Univariate Statistics

There are two formulae for computing the sample mean,

$$\bar{x}_1 = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

$$\bar{x}_2 = \frac{\sum_{i=1}^n x_i}{n} + \frac{\sum_{i=1}^n (x_i - \bar{x}_1)}{n} \quad (2)$$

If computation is exact (there is no rounding error) then the second term on the right-hand side of (2) equals zero and (2) collapses to (1). In the presence of rounding error, (2) can be expected to be more accurate than (1). Similarly, there are two formulae for computing the sample variance (the “calculator formula” is not even considered)

$$V_1 = \sum_{i=1}^n (x_i - \bar{x}_1)^2 / (n - 1)$$

$$V_2 = \frac{1}{n - 1} \left\{ \sum_{i=1}^n (x_i - \bar{x}_1)^2 - \frac{1}{n} \left[\sum_{i=1}^n (x_i - \bar{x}_1) \right]^2 \right\}$$

$$s_1 = \sqrt{V_1}, \quad s_2 = \sqrt{V_2}$$

The formula used by NIST to compute the first-order autocorrelation coefficient is

$$r_1 = \frac{\sum_{i=2}^n (x_i - \bar{x}_1)(x_{i-1} - \bar{x}_1)}{\sum_{i=1}^n (x_i - \bar{x}_1)^2} \quad (3)$$

There is more than one formula for computing the first-order autocorrelation (depending upon whether the full sample is used to compute the mean of the lagged series, etc.). Therefore, it is important to verify the specific formula used by a package.

These formulae were programmed in Lahey Fortran 90 on a WinTel machine using double precision (KIND=8). To measure the number of digits of accuracy in a computed quantity, x , compared to the benchmark value c , the log relative error (LRE) is used

$$\lambda = -\ln_{10}[|x - c|/|c|]$$

The following results were obtained, where the level of difficulty (low, average, high) is indicated next to the dataset name:

dataset	\bar{x}_1, \bar{x}_2	s_1, s_2	r_1
Pidigits (l)	15	15	15
Lottery (l)	15	15	15
Lew (l)	15	15	15
Mavro (l)	15	13.1	15
Michelso (l)	15	13.8	15
Numacc1 (l)	15	15	15
Numacc2 (a)	15	15	15
Numacc3 (a)	15	9.5	15
Numacc4 (h)	15	8.3	15

Table 1: StRD Results for Univariate Summary Statistics

For these datasets, there is no difference between the various methods; hence \bar{x}_1 and \bar{x}_2 share a column, and similarly for s_1 and s_2 . From this alone, it should not be concluded that formulae for \bar{x}_2 and s_2 are without merit. In particular, for large datasets \bar{x}_1 and s_1 may not be sufficiently accurate. [N.B. – the calculator formula delivers 2.5 digits for Numacc3 and zero digits for Numacc4.]

3 Analysis of Variance

The StRD datasets are all one-way balanced layouts. Even for such a simple design, there is a large variety of ways in which the problem can be structured. For example, in SPSS the problem can be given to four different commands: ONEWAY, ANOVA, MANOVA, and

dataset	reg.	reg. + i.f.
SiResist (l)	13.2	13.2
Simon1 (l)	14.8	15
Simon2 (l)	13.4	13.4
Simon3 (l)	12.2	12.4
AgWt (a)	11.7	11.7
Simon4 (a)	10.4	10.4
Simon5 (a)	10.2	10.2
Simon6 (a)	10.2	10.2
Simon7 (h)	4.4	4.4
Simon8 (h)	6.3	4.2
Simon9 (h)	0	4.2

Table 2: StRD results for ANOVA problems: regression and regression plus one step of iterative refinement

GLM; which implies four different structures. Once the problem is structured, there is also a large variety of ways in which it can be solved. For example, if the problem is cast in a regression framework, it can be solved via any of the various LU, QR, or SV decompositions. There are so many possible numerical solutions that, no effort is made to investigate to the different possibilities.

For present purposes, one method of casting the problem and one method of solution are adopted and discussed: a generic linear regression routine with (0, 1) dummy variables. In the usual fashion, the regression residual sum of squares is “within”, and this quantity subtracted from the response sum of squared deviations gives “between”. The particular regression routine employed delivers 8.3 digits for Longley (compared to 13.0, 12.1, and 8.6 for S-PLUS, SPSS, and SAS, respectively), so it is not the most accurate of available packages. Since any error in the standard ANOVA table will affect the F -statistic, it is the LRE of this statistic upon which attention centers.

Table 2 presents two sets of results for the ANOVA datasets. The first column gives dataset names and level of difficulty. The second column is straightforward solution by linear regression. The effect of cumulative rounding error quickly degrades the quality of the solution. This degradation is particularly pernicious since the software typically gives no indication of the extent of the rounding error. [There do exist procedures that provide error bounds and backward error estimates for the solution of such a problem (*e.g.*, LAPACK’s SGERFS subroutine), but such features typically are not provided in statistical software packages.]

The third column presents the same solution with one step of iterative refinement (a second step had no effect in all cases). Similar results were achieved by

recentering the response variable, which may be easier than iterative refinement. The sole anomaly is Simon8, where one step refinement has less accuracy than the original problem. This anomaly can be attributed to fortuitous cancellation.

It is interesting to compare simple regression with regression and one step of iterative refinement by examining their respective ANOVA tables for one problem, SmnLsg09.

StRD				
	df	ss	ms	F
bet.	8	160.0800	20.0100	2001.0000
w/in	18000	180.0000	0.0010	–
regression				
	df	ss	ms	F
bet.	8	150.5886	18.8232	1787.7322
w/in	18000	189.5234	0.0105	–
regression plus one step i.f.				
	df	ss	ms	F
bet.	8	160.0995	20.0124	2001.1349
w/in	18000	180.0098	0.0100	–

Table 3: StRD Results for ANOVA dataset SmnLsg09

It can be seen that the improvement in the accuracy of the solution can be attributed solely to the refinement of the regression residual sum of squares (which, as noted, can also be achieved by recentering the response variable). A procedure which separately calculated between and within sums of squares might be able to refine each separately, and thus achieve even greater accuracy.

Many of the ANOVA datasets, even the average difficulty ones on which many packages have delivered zero digits of accuracy, have a very small coefficient of variation. Thus, the objection has been raised that such datasets are unrealistic, and users will never encounter such problems. Murphy’s Law notwithstanding, this objection misses the larger point of benchmarking: What performance can a decent implementation of a decent algorithm achieve? If the answer to that question is six digits of accuracy (though a good implementation of a good algorithm might achieve, say, ten), then there is simply no reason for an algorithm that delivers zero digits of accuracy.

4 Linear Regression

Statistical packages solve linear regression problems many ways, including the Cholesky (CH), LU, QR, and SV decompositions. It is rumored that some packages

dataset	CHOL	QR	SVD
Norris (h)	12.0	11.9	7.5
Pontius (h)	11.2	13.7	12.7
Origin1 (h)	14.7	14.7	14.7
Origin2 (h)	15	15	15
Filip (h)	ns	7.4	6.3
Longley (h)	7.6	7.5	7.5
Wampler1 (h)	6.4	9.6	9.2
Wampler2 (h)	9.8	13.7	10.4
Wampler3 (h)	6.4	8.9	8.9
Wampler4 (h)	6.4	9.0	9.0
Wampler5 (h)	6.4	7.2	6.9

Table 4: StRD Results for Linear Regression using LAPACK

even employ direct solution of the normal equations via matrix inversion. The StRD datasets are solved using the LAPACK 77 versions of these algorithms (in particular, the Fortran-90 interface was not used). For QR and SV, the DGELS and DGELSS subroutines are used, respectively. For the Cholesky decomposition, the BLAS subroutines DGEMM and DGEMV are used to produce $X'X$ and $X'y$, respectively, and the system is solved using DPOSV. Results are presented in Table 4.

The Filip dataset merits brief discussion, for it is a tenth-order polynomial and nearly singular. Filip is the problem which most often has exposed flaws in linear regression routines. Reporting no solution (“ns”) is *not* a flaw, but indicates that the solver works well: it does not attempt to provide a solution to a problem which is too collinear. Reporting an inaccurate solution *is* a flaw, and indicates that the software either does not check for near singularity of the design matrix, or does a poor job of checking. To belabor this important point: software does not fail when it produces no solution; it fails when it produces an inaccurate solution.

An objection raised against this suite of tests, and Filip in particular, is that the datasets are unrealistic. Users in economics/sociology/biometrics/etc. will *never* regress a tenth order polynomial. Again, such an objection misses the larger point of benchmarking. A reliable linear solver should be able to diagnose near-singularity to prevent “solutions” completely corrupted by cumulated rounding error. The only way to test whether a linear solver has this necessary feature is to give it a nearly singular problem.

Additionally, the Filip problem has more than one solution, and persons who apply the StRD should be aware of this. The LAPACK routines did not find the StRD solution but another solution which produces the same residual sum of squares. The StRD solution has

ten negative coefficients; for the LAPACK solution, the magnitudes of the coefficients were the same but the signs alternated, with coefficients 2,4,6,8, and 10 being positive (a similar multiplicity of solutions occurs for some of the nonlinear problems). Observe in Table 4 that the QR never produces fewer accurate digits than the SVD, and usually provides more. This interesting phenomenon merits closer examination.

5 QR vs. SVD

To verify this interesting result, the datasets were solved again using the LU, QR and SVD routines from the S-PLUS (4.5) matrix library. Results are presented in Table 5 and again, the QR generally returns more accurate digits than the SVD.

dataset	LU	QR	SVD
Norris (h)	12.1	12.6	12.8
Pontius (h)	11.2	12.1	6.2
Origin1 (h)	14.7	14.7	14.7
Origin2 (h)	15	15	15
Filip (h)	0	8.4	5.8
Longley (h)	7.4	10.9	11.0
Wampler1 (h)	6.1	9.7	9.1
Wampler2 (h)	9.3	12.2	10.5
Wampler3 (h)	6.1	9.8	9.1
Wampler4 (h)	6.1	7.4	7.4
Wampler5 (h)	6.1	5.4	5.4

Table 5: StRD Results for Linear Regression from the S-PLUS matrix library

The differences between the S-PLUS and LAPACK results for QR, and for SVD, may be due to algorithmic differences. Returning again to consideration of the Filip dataset, observe that the Cholesky decomposition correctly reports “ns”, which is what a reliable linear solver should do when confronted with a problem that is too nearly singular for it to handle. By contrast, the particular LU procedure implemented in the S-PLUS matrix library does not recognize that the problem is too nearly singular, and unreliably returns an inaccurate answer.

Since rounding errors are highly nonlinear and the results in Tables 4 and 5 might be idiosyncratic, to investigate this matter more fully, a simple experiment was conducted. If b is the least squares solution to $y = X\beta$, then it is also the solution (at least in theory) to $\alpha y = (\alpha X)\beta$ where α is a scalar. Let ε be single precision machine epsilon. The scaling factor α is drawn from a uniform distribution on $[\varepsilon, 1/\varepsilon]$. Since computa-

dataset	QR	SVD	Wilcoxon	t -test
Norris (h)	12.5	12.6	6.5	5.2
Pontius (h)	12.2	6.3	-27.4	-301.1
Origin1 (h)	14.7	14.7	0.7	0.6
Origin2 (h)	14.7	14.7	0.6	-0.6
Filip (h)	7.5	5.7	-27.4	-104.7
Longley (h)	11.2	11.1	-7.1	-8.2
Wampler1 (h)	9.5	9.2	-16.7	-18.7
Wampler2 (h)	12.9	10.4	-27.4	-182.8
Wampler3 (h)	9.4	9.2	-16.3	-17.9
Wampler4 (h)	8.0	8.0	-1.6	-1.4
Wampler5 (h)	6.0	6.0	-0.4	1.4

Table 6: StRD Mean Results for Linear Regression from the Scaled Regressions

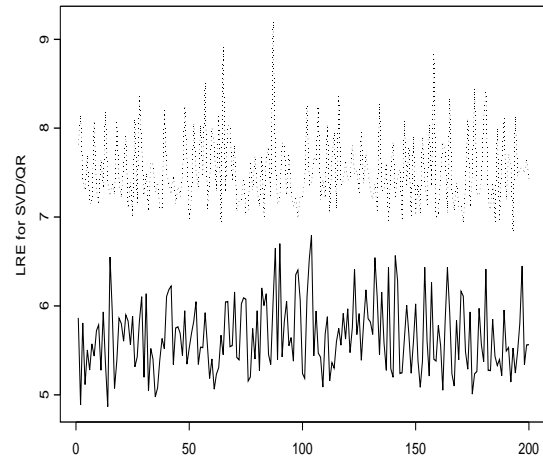


Figure 1: Filip experiment, draws 1-200; SVD (solid), QR (dash)

tion is in double precision, this interval provides a wide range without encountering the numerical difficulties associated with very small or very large α . One thousand times α is drawn, the scaled regression is solved using QR and SVD from the S-PLUS Matrix Library, and the LREs are computed. This experiment is repeated for each of the eleven linear regression datasets. The mean LREs for QR and SVD are given in Table 6, along with paired Wilcoxon and Student’s- t results to test whether the median/mean difference is zero.

Figure 1 displays the results of the first 200 draws for Filip. Clearly, for this dataset the QR appears to provide uniformly more accurate results than the SVD. Figure 2 shows the results for Longley. Even though the difference is statistically significant, for this dataset it appears not be practically important.

While it seems that there is no formal error analy-

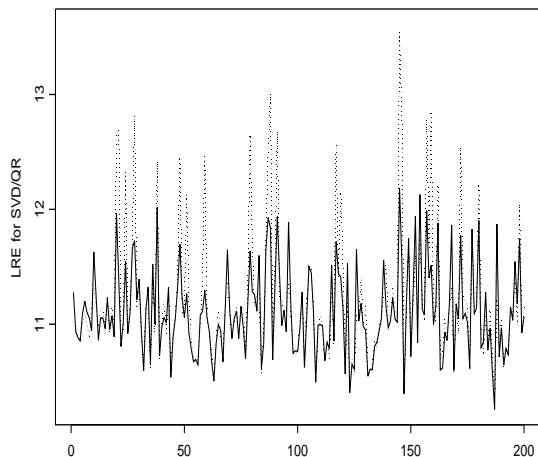


Figure 2: Longley experiment, draws 1-200; SVD (solid), QR (dash)

sis comparing the two algorithms, there is an intuitive explanation. The QR does significantly less arithmetic than the SVD, and is less likely to be corrupted by cumulated error. Therefore, these results in no way contravene the long-standing notion that the SVD is the regression procedure of choice. However, it does suggest that if a high degree of accuracy is needed, that the SVD first be used to determine whether the problem is amenable to treatment by less robust methods, such as the QR.

6 Nonlinear Least Squares

When confronted with a problem, a nonlinear solver can respond in one of three ways: (1) it can return an accurate solution; (2) it can report no solution (“ns”); or (3) it can produce a “solution” which has zero accurate digits. Reliable software will produce (1) or (2). It is important to recognize that (2) *is* a reliable answer, and not count such a response negatively. Consider, for example, Newton-Raphson (NR) and Gauss-Newton (GN). The former converges more quickly, but the latter is not as dependent upon “good” starting values. Suppose NR and GN are both applied to the same problem. That NR returns “ns” and GN returns an accurate solution does not imply that GN is “better” than NR or that NR is somehow deficient.

The 27 StRD nonlinear datasets come with two sets of starting values: Start I is “far” from the solution and Start II is “near” to the solution. For some problems, Start I might be quite far, and some solvers might not be able to find the solution from that point. As noted

above, this is not a cause for great concern. What is important is that the solver can recognize when it has not found a solution; *i.e.*, that it returns “ns” rather than a solution with zero accurate digits. Experience shows that many solvers are particularly poor at this important task. In this sense, it is a good thing that Start I is occasionally very far from the solution, for it gives the solver a chance to show that it can recognize when it cannot find a solution. Indeed, many packages perform admirably when given only the Start II solutions, but routinely produce zero digits of accuracy from Start I. From a practical perspective, Start I makes more sense than Start II. Frequently a user has little idea of what good starting values are, and is more likely to pick values far from the solution than close to it. In such a situation, it is important that a solver be able to recognize that it cannot find a solution.

In fact, since the StRD examines only an infinitesimal fraction of possible problems, and since no one algorithm is appropriate for all problems, it is important to maintain a proper perspective, and not place undue emphasis on Start I solutions at the expense of reliability. For example, one package only solved half the problems accurately, returning “ns” for the other half, while another package returned correct solutions for all but one, the one being inaccurate. The former package is more reliable, since it has not misled the researcher. On the other hand, if two packages both claim to implement GN and one typically solves from Start I while the other cannot, the latter package might be judged inadequate.

For some datasets, the StRD solution is not the only solution. To see this, consider the problem MGH17, which equation is

$$y = \beta_1 + \beta_2 \exp[-x\beta_4] + \beta_3 \exp[-x\beta_5] + \epsilon$$

Let the vector $\hat{b} = [b_1, b_2, b_3, b_4, b_5]$ be the StRD solution. By inspection, $\tilde{b} = [b_1, b_3, b_2, b_5, b_4]$ also is a solution. A solver might find \tilde{b} instead of \hat{b} , and should not be penalized for doing so. The Lanczos and Gauss datasets offer similar potential for multiple solutions.

Careful attention should be paid to coding the nonlinear equations. The StRD gives them in standard Fortran format, but many packages do not follow Fortran conventions. One common example is precedence of unary negation over exponentiation. Consider “ $y = -x * 2$ ”, and suppose $x = 2$. In Fortran, y evaluates to -4 , but in a package in which unary negation takes precedence over exponentiation, y evaluates to $+4$. For such a package, the Gauss problems must have extra pairs of parentheses inserted into the appropriate places in the equations. As an example, in Gauss1, the term

$b3 * \exp(-(x - b4) ** 2 / b5 ** 2)$ should be rewritten as $b3 * \exp(-((x - b4) ** 2) / b5 ** 2)$.

The StRD has been applied both formally and informally to over a dozen of the most popular statistics and econometrics packages, and only two of them have not produced nonlinear “solutions” with zero digits of accuracy: the statistical package S-PLUS (with analytic derivatives) and the econometrics package TSP (which has automatic analytic differentiation). S-PLUS solved all the problems, but required two different algorithms to achieve this; one of the solutions was from Start II, all attempts to solve from Start I having returned “ns”. TSP returned one “ns” and two solutions from Start II, but has only a single algorithm. No package employing numerical derivatives has even approached such reliable performance. At present, it appears that there is no one commercial package which will solve all 27 problems from Start I using a single algorithm.

Of course, the objection that these problems are unrealistic has been raised. So, the objection continues, a package’s performance on these problems is not a good indicator of the package’s true ability. This objection is utterly without merit. Whenever a nonlinear solver is given a problem it cannot handle, it should return “ns”.

7 Conclusions

The StRD has had an immediate and dramatic impact on commercial software for statistical and econometric packages. Of the seven packages assessed by McCullough (1999, 1999a), six have since come out with subsequent versions. Of those six, five developers report improved performance, and the sixth expects to do likewise with its next release.

The StRD has many additional uses in the context of assessing statistical software. Until more general benchmarks are available, the ANOVA datasets, which are for one-way balanced design, can be used to test special cases of more general procedures such as MANOVA or GLM. Similarly, until more general benchmarks are available, the linear regression datasets can be used to test special cases of GLM, GLS, and other procedures. The nonlinear least squares datasets can be applied (with null constraints) to the constrained nonlinear least squares procedure, and also to the more general unconstrained and constrained optimization routines.

Benchmarking requires an enormous amount of work, and users should not have to benchmark their packages. Neither should they have to wait until some reviewer does it. The software developer should provide tangible

evidence of reliability, including those items mentioned in the previous paragraph. Developers, however, being rational and profit-maximizing entities, will not supply this evidence unless users demand it. Therefore, users should contact their developers and inquire after tangible evidence of reliability, not just with respect to the StRD, but also for statistical distributions and the random number generator (McCullough 1998).

Interpreting such benchmark output, PC users would do well to keep in mind two important points. First, software packages should not be judged against the NIST results, but against the limits of 32 bit double precision. Second, when a procedure returns “ns” for a problem, this *is* evidence of reliability and, barring further detailed investigation, should not be construed as evidence of unreliable software.

ACKNOWLEDGEMENTS: Thanks to R. Beardsley, M. Keating and C. Romine for useful suggestions.

REFERENCES

- Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorenson (1995), *LAPACK User’s Guide, 2e*, Philadelphia: SIAM
- McCullough, B. D. (1998), “Assessing the Reliability of Statistical Software: Part I,” *The American Statistician*, **52**(4), 358-366
- McCullough, B. D. (1999), “Assessing the Reliability of Statistical Software: Part II,” *The American Statistician*, **53**(2), 149-159
- McCullough, B. D. (1999a), “Econometric Software Reliability: E-VIEWS, LIMDEP, SHAZAM, and TSP,” *Journal of Applied Econometrics*, **14**(2), 191-202
- McCullough, B. D. and Berry Wilson (1999), “On the Accuracy of Statistical Procedures in Microsoft EXCEL 97,” *Computational Statistics and Data Analysis*, **31**(1), 27-37
- Rogers, Janet, James Filliben, Lisa Gill, William Guthrie, Eric Lagergren and Mark Vangel (1998), “StRD: Statistical Reference Datasets for Testing the Numerical Accuracy of Statistical Software,” NIST# 1396, National Institute of Standards and Technology
- Vinod, H. D. and P. Silverio (1999), “A Review of GAUSS for Windows, Including Its Numerical Accuracy,” *Journal of Applied Econometrics*, forthcoming