

## The Devil is in the Detail: Hints for Practical Optimisation – A Comment

*B. D. McCullough*  
*Department of Decision Sciences*  
*Drexel University.*  
*(email: bdmccullough@drexel.edu)*

The recent paper by Christensen, Hurn and Lindsay (2008), ‘The Devil is in the Detail: Hints for Practical Optimisation’ offers some useful advice for practitioners. However, on one important point their advice could be improved. In the introduction to their paper, they write (p. 345), ‘At a superficial level, solving the minimisation problem is straightforward, usually involving only the choice of preferred econometric programming language like MATLAB, GAUSS, OX or RATS, or more interactive software packages like EViews. *These packages usually provide reliable default settings for the user-supplied information required by a minimisation routine.*’ [emphasis added] Nothing could be farther from the truth, and for two reasons: first, default options for a nonlinear solver are not likely to produce a correct answer; second, the answer produced by a nonlinear solver is not necessarily correct and needs to be verified by the user. This comment addresses both these points.

In general, the default settings will be wrong for whatever problem is given to the solver. For most any nonlinear problem, it is easy to see this simply by varying the default options and observing an improvement in the objective function. Mc-Cullough (1999a) tested four econometric software packages on the 27 nonlinear problems from the NIST ‘Statistical Reference Datasets’ (StRD) and found that none of them could solve all the problems at default. In fact, one package could only solve 13 of the 27 problems at default. McCullough (1999b) performed similar tests on three statistical software packages, and found that all failed to solve six or seven problems at default. Other researchers testing software using the StRD have found similar results, *e.g.*, Vinod (2000), Keeling and Pavur (2007), Yalta (2007).

There are two ways a solver may fail to solve a nonlinear problem (Murray, 1972, p. 107): ‘The first is miserable failure which is discovered when an exasperated computer finally prints out a message of defeat. The second is disastrous failure when the computer and trusting user mistakenly think they have found the answer.’ Unfortunately, when a solver employing default (or even non-default) options produces an ‘answer’ it is often a disastrous failure: a numerical answer that has no correct digits. All of the packages analyzed by McCullough (1999a, 1999b) produced disastrous failures.

One thing we know from the literature on testing nonlinear solvers is that it is not safe to trust default options. Therefore, the user should experiment with the options (*e.g.*, switch algorithm from BHHH to BFGS, tighten the convergence tolerance from  $1 \times 10^{-3}$  to

$1 \times 10^{-7}$ , vary the termination criterion, etc.), and see what happens to the objective function as recommended in McCullough and Renfro (1999). However, even changing the options is insufficient to ensure that a correct solution has been found.

For example, in Professor Robert Engle's 'GARCH 101 Tutorial' he presented estimates of a GARCH(1,1) model based on running a nonlinear solver at default and obtained the following set of results:  $\mu = 1.40 \times 10^{-6}$ ,  $a_1 = 0.0772$  and  $\beta_1 = 0.9046$ . Simply tightening the convergence tolerance from the default  $1 \times 10^{-3}$  to  $1 \times 10^{-7}$  changed the results to  $\mu = 1.09 \times 10^{-6}$ ,  $a_1 = 0.0654$  and  $\beta_1 = 0.9202$ . See McCullough and Vinod (2003b) for further discussion of this example. McCullough (2004) gives an extended example in which a solver gives a different answer every time the convergence tolerance is changed, and many of the answers are completely wrong. Stokes (2004) gave the same problem to several packages and got several different answers. Which of the many answers was correct?

In fact, when a solver produces a 'solution' the researcher's task is just beginning. How can a user verify that the 'solution' offered by a nonlinear solver is, in fact, correct? To answer this question, McCullough and Vinod (2003a) offered a four step procedure for verifying the solution from a nonlinear solver.

- Step 1: Examine the gradient – is it zero?
- Step 2: Inspect the trace – does it exhibit the characteristics indicative of successful convergence?
- Step 3: Evaluate the Hessian – is it positive definite (for minimization)?
- Step 4: Profile the likelihood – is the surface approximately quadratic?

The interested reader is referred to McCullough and Vinod (2003a) for further details and an application. Note that this article made an error in step 3, which was noted by Drukker and Wiggins (2004) and corrected in McCullough and Vinod (2004). Specifically, before concluding that a Hessian is ill-determined, one must be careful to distinguish between artificial ill-conditioning and inherent ill-conditioning. A further application of this four-step method is presented in McCullough (2004).

The paper by Christensen, Hurn and Lindsay does a fine job of explaining how to use a nonlinear solver. But the user needs to remember to experiment with the default options to find a tentative solution, and then employ the four-step method to verify that the candidate solution is a true solution. An introduction to the numerical issues bedeviling econometric software – beyond nonlinear solvers can be found in McCullough and Vinod (1999).

#### REFERENCES

- Christensen, T.M., A.S. Hurn, and K.A. Lindsay (2008). The Devil is in the Detail: Hints for Practical Optimisation, *Economic Analysis and Policy*. 38: 345-368.
- Drukker, D.M. and V. Wiggins (2004). Comment, *American Economic Review*. 94: 397-399.
- Engle, R.F. (2001). GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics, *Journal of Applied Econometrics*. 15: 157-168.
- Keeling, K.B. and R.J. Pavur (2007). A comparative study of the reliability of nine statistical software packages, *Computational Statistics and Data Analysis*. 51: 3811-3831.
- McCullough, B.D. (1998). Assessing the Reliability of Statistical Software: Part I, *The American Statistician*. 52: 358-366.

- McCullough, B.D. (1999). Assessing the Reliability of Statistical Software: Part II, *The American Statistician*. 53: 149-159.
- McCullough, B.D. (2004b). Some Details of Nonlinear Estimation, Chapter Eight in: Micah Altman, J.G. and M.P. McDonald (eds.), *Numerical Methods in Statistical Computing for the Social Sciences*. New York: Wiley.
- McCullough, B.D. and C.G. Renfro (2000). Some Numerical Aspects of Nonlinear Estimation, *Journal of Economic and Social Measurement*. 26: 63-77.
- McCullough, B.D. and H.D. Vinod (1999). The Numerical Reliability of Econometric Software, *Journal of Economic Literature*. 37: 633-665.
- McCullough, B.D. and H.D. Vinod (2003a). Verifying the Solution from a Nonlinear Solver: A Case Study, *American Economic Review*. 93: 873-892.
- McCullough, B.D. and H.D. Vinod (2003b). Comment: Econometrics and Software (with H. D. Vinod), *Journal of Economic Perspectives*. 17: 223-224.
- McCullough, B.D. and H.D. Vinod (2004). Verifying the Solution from a Nonlinear Solver: Reply, *American Economic Review* 94: 400-403.
- Murray, W. (1972). Failure, the Causes and Cures, in: W. Murray (ed.) *Numerical Methods for Unconstrained Optimization*. New York: Academic Press, p. 107-122.
- Stokes, H.H. (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.
- Vinod, H.D. (2000). Review of GAUSS for Windows, including its numerical accuracy, *Journal of Applied Econometrics*. 15: 211-222.
- Yalta, T. (2007). Numerical Reliability of Econometric Software: The Case of GAUSS 8.0, *The American Statistician*. 61: 262-268.

