Chapter 10
NONLINEAR MINIMIZATION METHODS AND CONVERGENCE OPTIONS

The estimation procedures described in the previous chapters -- LSQ, FIML, PROBIT, TOBIT, SAMPSEL, LOGIT, and ML, involve iterative methods. In this chapter, we describe the methods used to converge to a solution. Many options are available in the event that convergence is difficult to obtain, while other options allow the calculation of alternate standard errors for the estimated parameters.

10.1. Nonlinear minimization methods for estimation

All of TSP's nonlinear statistical techniques involve the minimization of a criterion function, Q, over the parameters. For single equation least squares the criterion is the sum of squared residuals, while other estimators use more complicated criteria such as (minus) the log of the likelihood function. The minimization strategy is always the same and we give a brief summary of it here. In general, the methods used are gradient methods and make use of analytic derivatives of the model obtained internally by TSP. The user may request numeric derivatives with the GRAD= option (see the *Reference Manual* for details).

Consider Q as a function of the parameters b: Q(b). Let d be a vector with the property that Q decreases in the direction defined by d. In general, d is approximately equal to $H^{-1}g$, where g is the gradient of the function Q and H an approximation to the Hessian matrix (the second derivatives of Q with respect to b). Q(b+λ*d) will be a decreasing function of the scalar, λ, at least for very small values of λ, since g is a gradient and H is positive definite. TSP proceeds iteratively in the following way: at the beginning of iteration i, parameter values b^{i-1} are available. TSP computes the change vector d from g and H. Next it checks for convergence, defined as sufficiently small elements of d. For each parameter, |d| must be less than a prescribed tolerance, TOL, or $|d/b^{i-1}|$ must be less than the same tolerance. The exact formula used is

$$|d| \le TOL*(|b^{i-1}| + TOL*10)$$

If convergence has not been achieved, TSP goes on to find a better set of parameters, b^i. It first tries

$$b^i = b^{i-1} + d$$

If b^i is better, i.e., if

$$Q(b^i) < Q(b^{i-1}),$$

the iteration is complete and the next one begins with the parameter b^i. Otherwise, TSP begins to try parameter vectors of the form

$$b^i = b^{i-1} + \lambda*d$$

first for the stepsize λ = 1/2, then λ = 1/4, and so on. This process is called "squeezing" and continues until a better b^i is found or a squeeze limit is exceeded. If squeezing is successful, TSP goes on to a new iteration. If it fails, iteration stops and a message is printed. The whole iteration process continues until convergence, or until the iteration limit, MAXIT, is reached. Results are printed whether or not convergence has been achieved, since the process is often close to convergence (how close can be seen by examination of the last value of CRIT =, which is approximately equal to the average error in the parameters squared). These results need to be interpreted with care, and are statistically meaningless if the process is far from convergence.

10.2. General convergence hints

Several options control this iteration process, and may be useful for reaching convergence in highly nonlinear or barely identified models. In general, starting values are most critical for success with difficult problems, but iteration methods

may help somewhat. Starting values obtained from alternate consistent estimators, such as those based on smaller incomplete models are usually useful. An example would be to use single equation estimates before estimating all the equations jointly. Problem parameters are identified by checking the CHANGES (d) row in the output for especially large values. These parameters can be held fixed by use of the CONST statement, reducing the dimension of the parameter vector until the remaining parameters have converged. Then the problem parameters can be re-entered with a PARAM statement, possibly with additional experimentation in their starting values. Large CHANGES values can also indicate lack of identification in the current data set.

Grid searches can be performed to obtain starting values for critical parameters. DO loops are ideal for this purpose, since the index and step variables need not be integers (see Section 12.1 for more information on DO). For example, AR1(METHOD=HILU) Y C X; could be done with:

```
FRML EQA Y = A + B*X + RHO*(Y(-1)-A-B*X(-1));
PARAM A,B;
DO RHO = -.95 TO .95 BY .1;
  LSQ EQA;
ENDDO;
```

Iteration options can also help obtain convergence for some models. These options are explained in detail below; they are also discussed in the NONLINEAR section of the *Reference Manual.*

10.3. Diagnostic printing: PRINT, VERBOSE, SILENT

The first step in tackling a difficult convergence problem is to insure you are getting enough diagnostic output.

The SILENT option suppresses all iteration output, except a possible non-convergence message with full display of function values and changes in the last iteration. It is useful only when convergence is certain.

Slightly more output is provided by the default NOPRINT option, which displays the starting values. For each iteration, it prints the function values $F=Q(b^{i-1})$ and $FNEW=Q(b^{i})$, squeeze level (ISQZ, STEP or STEPSIZE=λ), and gradient norm (CRIT or CRITERION=$g' H^{-1} g$ or $d' Hd$). The gradient norm is the length of the gradient in the metric of the Hessian approximation. This should always decrease; otherwise, convergence will be difficult to obtain. At convergence, the number of iterations and function evaluations (Q for different values of b) is displayed.

The PRINT option includes all the above output, along with current parameter values and their changes (the d vector) at each iteration. The complete list of iteration options is also displayed at the start.

The VERBOSE option displays g, H, and H^{-1} at each iteration.

10.4. Numerical error handling

Some nonlinear equations involve functions sensitive to bad or unrestricted parameter values. For example, SQRT is defined only for non-negative values, LOG is defined only for positive values, and EXP typically causes problems for arguments larger than 88 (on most computers). Again, starting values are important, but some diagnostic output is provided to help solve the problem. When numeric errors are encountered, the offending observation number and argument value are printed for the first 10 occurences. If this occurs during a derivative evaluation, the number(s) of the parameter(s) involved are printed.

TSP requires that the starting values be good enough so that no numeric errors are encountered in the initial function and derivative evaluation, although it allows them during the stepsize search. A valid Q value is required for testing iterative improvement, while a valid d (i.e., g and H) is required for calculating b^{i}. Otherwise it is not possible to do any iterations. If there are outliers in the dataset, it may be useful to drop them from the sample (at least until better starting values are found).

During iterations, numeric errors in evaluating Q are not a problem, since λ is squeezed repeatedly until the numeric

errors stop. Eventually b^i will be close to the valid b^{i-1} if squeezed far enough. Numeric errors in derivatives are still a potential problem, but are less likely to occur since the derivatives are only evaluated when a valid and improved Q has been found.

If a problem requires restricting a parameter to a certain range, often the best solution is to enter the parameter in a functional form, making this restriction explicit. For example, if a parameter is restricted to be non-negative, it can be entered into the equation(s) as its square (D*D instead of just D). To restrict a parameter D to the range [a,b], use the function a+(b-a)*CNORM(D). ANALYZ can be used later to obtain a standard error for the entire function.

10.5. Hessian and gradient methods: HITER, HCOV

Iteration algorithm options give variations in computing H and g, the components of d. (For more information see Gill (1981).) The HITER=B or N or G or D option specifies the method of approximating H during the iterations. This may be different from HCOV=, which specifies the method of approximating H in computing the covariance matrix (and thus standard errors) for the parameters at convergence. The HCOV= option allows for printing more than one set of standard errors (and asymptotic t-statistics) for a given set of estimated coefficients, by specifying an option like HCOV=NBW instead of just HCOV=N. Depending on the estimation command (LSQ, FIML, PROBIT, ML, etc.), the available options for HITER= and HCOV= vary. Check the *Reference Manual* for further details.

HITER=B ("BHHH") specifies use of the Berndt-Hall-Hall-Hausman method. It uses the covariance of the analytic gradients for each observation to form H. It has the advantage of being easy to compute and is guaranteed non-negative definite as long as the number of observations is greater than the total number of parameters.

HITER=N ("Newton") uses analytic second derivatives to form H. This may be time-consuming to compute for nonlinear models, but provides faster convergence near the solution and HCOV=N usually yields the smallest estimate of standard errors.

HITER=G ("Gauss") is standard for LSQ and FIML, and involves a quadratic form in the derivatives of the residuals with respect to the parameters, around the estimated residual covariance matrix.

HITER=D ("DFP") uses the Davidon-Fletcher-Powell method to update H at each iteration based on the gradient and parameter changes (H is started as an identity matrix). It also implies the use of numeric derivatives to compute g. Analytic derivatives are the default method for computing g. HITER=D is only useful in the most extreme cases, for bad starting values, because it is very slow. The SYMMETRI option specifies use of the most accurate (and time-consuming) numeric derivatives.

HITER=F (“BFGS”) uses the Broyden-Fletcher-Goldfarb-Shanno algorithm with analytic first derivatives and a rank 1 update approximation to the Hessian (like DFP, but somewhat improved).

There are two HCOV= options that are not used for HITER= :

HCOV=W ("Eicker-White") computes standard errors based on a combination of the BHHH and Newton matrices. Useful for some forms of misspecification in maximum likelihood estimation.

HCOV=R ("Robust") computes standard errors robust to heteroskedasticity. Used in LSQ only, this is equivalent to the old ROBUST option, and it uses a formula equivalent to the W option.

10.6. Squeezing: STEP, MAXSQZ

The algorithm that generates successive values of the stepsize k is chosen by the STEP option. The default STEP method depends on the HITER option and on the estimation procedure. The available methods are CEA, BARD, CEAB, BARDB, and GOLDEN.

STEP=CEA is the simplest method. λ = 1, .5, .25, ..., 2**(-ISQZ). This is the default for HITER=N and

HITER=B (procedures like PROBIT and ML).

STEP=BARD uses a local quadratic approximation to the function value based on the previous value of λ. It is also bounded in the interval $[.75\lambda,.25\lambda]$. A typical sequence could be λ = 1, .25, .0625, ... This is the default for LSQ with HITER=G.

STEP=CEAB is the same as CEA, but if λ = 1 improves the objective function, λ = 2, 4, ... etc. are used to see if they result in further improvement. This method may be useful if λ = 1 is always improving the function, but only slowly, and the elements of d always have the same sign. This is FIML's default with HITER=G.

STEP=BARDB is the analogous extension to $\lambda > 1$ for BARD.

STEP=GOLDEN is a bracketing method that tries smaller and larger λ values even after a λ has been found which improves the objective function. The bracketing stops with the current best value of λ when MAXSQZ is reached or when λ has been determined up to the tolerance specified in the SQZTOL option. The default value of SQZTOL is .1 . GOLDEN is the default for HITER=D.

The MAXSQZ option limits the total number of λ values attempted in a given iteration. The default value of MAXSQZ is 10 for all STEP options except GOLDEN, where the default value is 20. If ISQZ > MAXSQZ, the message "FAILURE TO IMPROVE OBJECTIVE FUNCTION (MAXSQZ)" will appear. Increasing MAXSQZ will not automatically eliminate this problem; first check the CHANGES row to see which parameters are causing the trouble. Better starting values for those parameters may help.

10.7. Overall options: MAXIT, TOL

The MAXIT option specifies the maximum number of iterations. The default is 20 for all procedures. If MAXIT is exceeded, the message "CONVERGENCE NOT ACHIEVED AFTER n ITERATIONS" will appear. It may be useful to increase MAXIT if you have many parameters. However, if the CRIT value (section 10.3) is not decreasing smoothly, the objective function may be very non-quadratic near the current parameter values, so changing MAXIT may not help (again, better starting values may help).

The TOL option checks for convergence at the start of each iteration (section 10.1). The default value is .01 for all procedures. Usually there is little reason to increase TOL, except possibly to reduce the number of iterations in preliminary runs. TOL can be decreased if extremely accurate parameter values are desired.