Nonlinear Regression Analysis and Nonlinear Simulation Models

Donald Erdman and Mark Little, SAS Institute Inc., Cary, NC

Abstract

This paper is a survey of SAS System features for nonlinear models, with emphasis on new features for nonlinear regression. Topics include automatic calculation of analytic derivatives, estimation with nonlinear parameter restrictions, tests of nonlinear hypotheses, maximum likelihood and generalized method of moments (GMM) estimation, estimation of simultaneous systems of nonlinear regression equations, and distributed lags and time series error processes for nonlinear models. In addition, this paper will briefly discuss solving nonlinear equation systems, dynamic simulation of nonlinear systems, and optimization of nonlinear functions. The MODEL, NLIN, NLP, and GENMOD procedures are discussed.

Introduction

For simplicity, many researchers assume that their problems can be represented by linear models. This assumption is valid if the problem is truly linear or if you are restricted to studying only a small area of the problem space. Other researchers transform their problems appropriately to obtain linear models. While linear models are useful for much research, nonlinearity pervades our every day life and should not be ignored.

This paper will concentrate on the estimation and simulation of nonlinear models. Estimation of nonlinear models usually requires finding the minimum (or maximum) of a nonlinear function. Solving for unknown variables in nonlinear equations requires finding zeros of the equations as a function of the unknown variables.

Nonlinear Estimation

A model can be nonlinear in its parameters, nonlinear in its observed variables, or nonlinear in both its parameters and variables. *Nonlinear* in the parameters means that the mathematical relationship between the variables and parameters is not required to have a linear form. (A linear model is a special case of a nonlinear model.)

Example of Nonlinear Estimation

Consider a simple exponential model for the decay of a radioactive isotope:

$$conc = conc_0 * \text{exp}(rate * t) \qquad (1)$$

where $conc_0$ is the initial concentration, t is time, and $rate$ is the rate of decay. This model can also be written as a linear model with a log *link function*, the function that associates the regressors with the response variable.

$$\text{log}(conc) = logconc_0 + rate * t \qquad (2)$$

Because the model can be written as a generalized linear model, the GENMOD procedure can be used to estimate the model parameters using the following SAS code:

```
proc genmod data=decay;
   model conc = t / dist = normal
            link = log noscale;
run;
```

The output is shown in Output 1. The reported INTERCEPT value of 1.3756 is the log of the parameter $conc_0$.

Output 1. PROC GENMOD Estimation Results

```
                           The SAS System


                        The GENMOD Procedure

                   Analysis Of Parameter Estimates

   Parameter    DF    Estimate     Std Err   ChiSquare  Pr>Chi

   INTERCEPT     1      2.2721      0.0516   1936.6188  0.0000
   T             1     -0.0568      0.0040    201.5282  0.0000
   SCALE         0      1.0000      0.0000           .       .
NOTE:  The scale parameter was held fixed.
```

This estimation can also be carried out by one of the more general purpose nonlinear estimation procedures. Figures 1 through 3 show how this same estimation is performed using the MODEL, the NLP, and the NLIN procedures.

```
proc nlp data=decay ;
   lsq z;
   parms conc_0 = 3, rate = -0.1;

   z = conc_0 * exp( rate * t) - conc;
run;
```

Figure 1. Least Squares Regression by PROC NLP

```
proc model data=decay;
   parms conc_0 3 rate -0.1;

   conc = conc_0 * exp( rate * t);

   fit conc ;
run;
```

Figure 2. OLS Regression by PROC MODEL

```
proc nlin data=decay;

   parms conc_0 = 3, rate = -0.1;

   model conc = conc_0 * exp( rate * t);

   der.conc_0 = exp( rate * t);
   der.rate = conc_0 * exp( rate * t) * t;

run;
```

Figure 3. OLS Regression by PROC NLIN

The output for the MODEL, NLIN, and NLP procedures is shown in Output 2 through Output 4. Why are the parameter values different after four decimal places ? Nonlinear estimation requires an iterative process to find the parameter estimates. Because each procedure uses a different iterative process and a different criterion for terminating this iterative process, the procedures will not, in general, produce exactly the same parameter estimates. To get more decimal places to agree, you can change the termination criterion on each routine to force them to do more iterations and get closer to the minimum. Remember, the standard errors and assumptions about the model still dictate the accuracy of the estimation.

The reported standard errors for the RATE parameter are much different for the GENMOD procedure than for the NLIN and MODEL procedures. The NLIN and MODEL procedures use the cross products approximation to the Hessian to estimate the standard errors. The GENMOD procedure uses the exact Hessian to estimate the standard errors.

Output 2. Proc NLP Estimation results

```
                              The SAS System

                 PROC NLP: Least Squares Minimization

Active Constraints= 0  Criterion= 11.3
Maximum Gradient Element= 0.004 Radius= 0.00201

NOTE:  GCONV convergence criterion satisfied.
NOTE: At least one element of the (projected) gradient is
      greater than 1e-3.

                      Optimization Results
                       Parameter Estimates
           ------------------------------------------
              Parameter      Estimate      Gradient
           ------------------------------------------
              1  CONC_0        9.699391  -1.087511E-9
              2  RATE         -0.056782      0.004003
```

Output 3. Proc MODEL Estimation results

```
                            The SAS System

                            MODEL Procedure
                            OLS Estimation

                   Nonlinear OLS Parameter Estimates

                                  Approx.       'T'   Approx.
      Parameter    Estimate      Std Err     Ratio  Prob>|T|

      CONC_0       9.699380      0.25633     37.84    0.0001
      RATE        -0.056781    0.0020644    -27.50    0.0001



       Number of Observations       Statistics for System
       Used                 90      Objective      0.2515
       Missing               0      Objective*N   22.6376
```

Output 4. Proc NLIN Estimation results

```
                            The SAS System

             Non-Linear Least Squares Summary Statistics
                       Dependent Variable CONC

Source                  DF Sum of Squares     Mean Square

Regression               2   782.24526667    391.12263333
Residual                88    22.63764764      0.25724600
Uncorrected Total       90   804.88291430

(Corrected Total)       89   508.93895487


Parameter     Estimate    Asymptotic             Asymptotic 95 %
                          Std. Error         Confidence Interval
                                             Lower         Upper
  CONC_0    9.699397393 0.25633012909  9.1899927365  10.208802050
    RATE   -0.056781581 0.00206443916 -0.0608842392  -0.052678923
```

Time Series Issues

Econometric models often explain the current values of the dependent variables as functions of past values of the dependent and independent variables. These past values are referred to as *lagged* values, and the variable x_{t-i} is called lag *i* of the variable x_t. Using lagged variables, you can create a *dynamic*, or time-dependent, model.

If the data are time series, so that *t* indexes time, it is possible that ϵ_t, the error of the model at time *t*, depends on ϵ_{t-i} or, more generally, the ϵ_t's are not identically and independently distributed. If the errors of a model are autocorrelated, the standard error of the estimates of the parameters of the system will be inflated.

Sometimes the ϵ_t's are not identically distributed because the variance of ϵ is not constant. This is known as *heteroscedasticity*. Heteroscedasticity in an estimated model can also inflate the standard error of the estimates of the parameters. Using a weighted estimation, you can sometimes eliminate this problem. If the proper weighting scheme is difficult to determine, generalized methods of moments (GMM) estimation can be used to determine parameter estimates.

Example of Time Series Estimation

Suppose you want to model the average monthly temperature in Cary, North Carolina. Using data from the local

National Weather Service, you can fit the following model using the MODEL procedure:

$$avgtemp_t = a * avgtemp_{t-12} + b * avgtemp_{t-6} + c; \quad (3)$$

This equation says that the average temperature for month *t* is linearly related to the average temperature a year ago and six months ago.

The following code was used to perform the estimation:

```
proc model data=gaspwr ;

  avgtemp = a * lag12(avgtemp) +
            b*lag6(avgtemp) + c;

  fit avgtemp / out=err;
run;
```

Figure 4. Time Series Regression Using PROC MODEL

For the estimation, the natural log of the average temperature was used. The output from the estimation is shown in Output 5.

Output 5. Time Series Regression Output

```
                         The SAS System

                        MODEL Procedure
                         OLS Estimation

           Nonlinear OLS Summary of Residual Errors

                 DF    DF
   Equation Model Error       SSE       MSE R-Square Adj R-Sq

   AVGTEMP      3    14    0.0293  0.002090   0.9692   0.9648


                         The SAS System

                        MODEL Procedure
                         OLS Estimation

              Nonlinear OLS Parameter Estimates

                                Approx.       'T'   Approx.
    Parameter    Estimate      Std Err      Ratio  Prob>|T|

    A            0.615083      0.20866       2.95    0.0106
    B           -0.503895      0.19924      -2.53    0.0241
    C            3.608752      1.64540       2.19    0.0457
```

If you look at the residual plot in Figure 5, the residuals do not seem independent. Using the new TESTEQ statement in the MODEL procedure, you can test for the heteroscedasticity of the residuals using either a modified Breusch-Pagan test or White's test. The null hypothesis for these tests is that the variances of the errors (ϵ_t) are equal. Using the Breusch-Pagan test and an $\alpha = 0.05$, you cannot reject the null hypothesis.

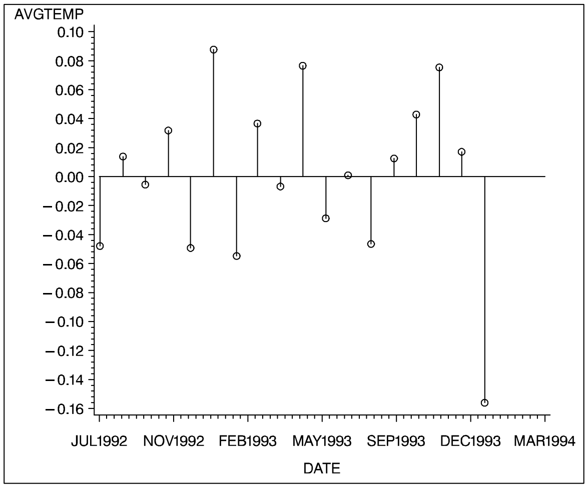

Figure 5. Residuals from OLS

Example Two of Time Series Estimation

The decay model is also a time series model, and its residual plot is shown in Figure 6.

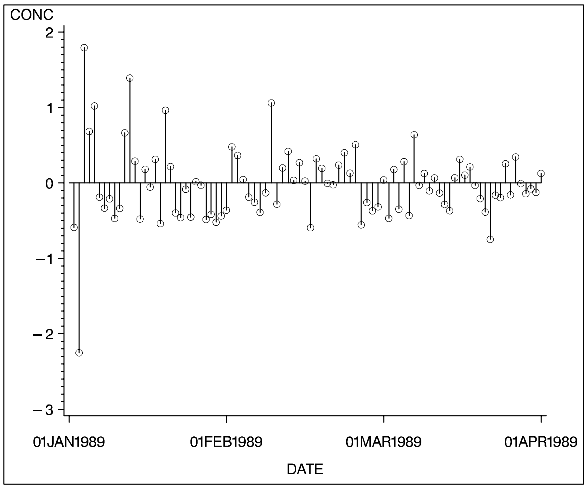

Figure 6. Residuals for Decay Model

These residuals are clearly heteroscedastic, and the Breusch-Pagan heteroscedasticity test supports this hypothesis.

The WEIGHT statement or _WEIGHT_ variable can be used to correct for heteroscedasticity. If you look at the residual plot in Figure 5, the first part of the data seems to have a larger variance than the last half of the data. For this example, if you use a simple weight like the observation number, you give less weight to the higher variance observations and more weight to lower variance observations. You can do this using the MODEL procedure with the following SAS statement after the FIT statement:

```
weight _OBS_;
```

The GENMOD, NLIN, and NLP procedures also support weighted regression.

The other alternative is to use Generalized Method of Moments (GMM). This is done by modifying the preceding FIT statement to:

```
fit avgtemp / out=err gmm;
```

The new set of estimates is shown in Output 6.

Output 6. GMM Time Series Regression Output

```
                         The SAS System

                         MODEL Procedure
                         GMM Estimation

            Nonlinear GMM Summary of Residual Errors

                DF    DF
    Equation Model Error       SSE       MSE R-Square Adj R-Sq

    CONC         2    88   22.8722   0.25991   0.9551   0.9545

                         The SAS System

                         MODEL Procedure
                         GMM Estimation

                Nonlinear GMM Parameter Estimates

                                Approx.       'T'   Approx.
      Parameter    Estimate     Std Err     Ratio  Prob>|T|

      CONC_0      9.943523     0.44290     22.45    0.0001
      RATE       -0.058180   0.0021786    -26.70    0.0001
```

Multiple Equation Systems

If a model has more than one dependent variable, you must be careful in choosing an estimation method. If the model has dependent regressors, then nonlinear ordinary least-squares estimation of these equations will produce biased and inconsistent parameter estimates. This is called *simultaneous equation bias*. For example, consider the following two-equation system:

$$y_1 = a_1 + b_1 y_2 + c_1 x_1 + \epsilon_1 \qquad (4)$$

$$y_2 = a_2 + b_2 y_1 + c_2 x_2 + \epsilon_2 \qquad (5)$$

In the first equation, y_2 is a dependent, or *endogenous*, variable. As shown by the second equation, y_2 is a function of y_1 and therefore y_2 depends on ϵ_1. Likewise, y_1 depends on ϵ_2 and is a dependent regressor in the second equation. This is an example of a *simultaneous equation* system; y_1 and y_2 are a function of all the variables in the system.

Two-stage least squares, or 2SLS, is used to avoid simultaneous equation bias. This is done with a set of variables called *instruments* which you select. The instrumental variables are used to filter out the error introduced by the dependent regressor. Refer to Amemiya (1985, p. 250) for details on the properties of nonlinear two-stage least squares.

When you have a system of several regression equations, the random errors of the equations can be correlated. In this case, the large-sample efficiency of the estimation can be improved by using a joint generalized least-squares method that takes the cross-equation correlations into account. If the equations are not simultaneous (no dependent regressors), then *seemingly unrelated regression* (SUR) can be used.

If the equation system is simultaneous, you can combine the 2SLS and SUR methods to take into account both simultaneous equation bias and cross-equation correlation of the errors. This is called *three-stage least squares* or 3SLS.

A different approach to the simultaneous equation bias problem is the full information maximum likelihood, or FIML, estimation method. FIML does not require instrumental variables, but it assumes that the equation errors have a multivariate normal distribution. 2SLS and 3SLS estimation do not assume a particular distribution for the errors. Note that the GENMOD procedure supports maximum likelihood estimation but cannot estimate systems of equations.

Example Multiple Equation Estimation

Using some relationships from physics, you can model a power (electric) bill and gas bill. For this model, assume that if you did not run your gas furnace or the air conditioner the power bill and gas bill would be constant from month to month.

First model the gas bill. The furnace runs to offset the heat flowing out of the house. The rate of heat flow out of your your house is dependent on the temperature difference between the outside and inside of your home, the amount of insulation in your home, and the number of kids you have who leave the door open when they come in and out of your home. From a first year physics book (Ignoring the effect of the children) that relationship is

$$rate\ of\ heat\ flow = A/R * \Delta T \qquad (6)$$

where R is the resistance (R-value) of the walls and ceiling of your home, A is the total area of the outside walls ceiling in square feet, and $\Delta T = (T_{outside} - T_{inside})$.

Air conditioning is more complex. As the temperature difference between the inside of the house and the outside of the house increases, the amount of time the air conditioner's compressor runs increases nonlinearly. Running the compressor is the majority of the cost of using an air conditioner. From a refrigeration engineer (Chrzanowski 1994), you can get an equation of the following form for the run time of a compressor as a function of temperature:

$$runtime = K(\Delta T)^2 / \exp(s \Delta T) \qquad (7)$$

where K and s are parameters that depend on the properties of the refrigerant and compressor, and ΔT is the temperature difference the air conditioner is working against.

Using the preceding two equations, you can model the gas and power consumption of your home as a function of outside temperature. If you are willing to neglect the effect of inflation on gas and power bills, you can relate the temperature outside directly to your gas and power bills. You will model the gas bill differently for the warm months than for the cold months. The gas bill for the warm months is modeled as a constant and for the cold months it is modeled as

$$gasbill = ga * (rate\ of\ heat\ flow) + gb \tag{8}$$

Which says that the gas bill is proportional to the heat flowing out of the house. You take a similar approach with the power bill. In the cold months, the power bill is considered constant, and in the warm months its modeled using:

$$powbill = pa * (runtime) + pb; \tag{9}$$

where *runtime* is the run time of the compressor. An SUR estimation is performed on a model for gas and power bills using the MODEL procedure.

```
proc model data=gaspwr outmodel=gasp;
   parms ga gb pa pb pc -0.7;
   control t_inside_winter 65
           t_inside_summer 70;

         /* ------- Model Gas Bill --*/
   if t_inside_winter < avgtemp then
      gasbill = gb;
   else
      gasbill = ga * (t_inside_winter - avgtemp)
                + gb;

         /* ------- Model Power Bill --*/
   dt = avgtemp - t_inside_summer;
   if t_inside_summer > avgtemp then
      powbill = pb;
   else
      powbill = pa* dt* dt* exp(pc * dt)
                + pb;

   fit powbill gasbill / sur
         method=marquardt;
run;
```

Figure 7. Gas/Power Regression by PROC MODEL

The actual local average monthly temperatures and the author's gas and power bills for the past three years were used for data for this model. A summary of the output is shown in Output 7.

Output 7. SUR Estimation results

```
                         MODEL Procedure
                         SUR Estimation

               Nonlinear SUR Summary of Residual Errors

                  DF    DF
   Equation Model Error       SSE       MSE R-Square Adj R-Sq

   POWBILL      3    25  847.9065  33.91626   0.9355   0.9303
   GASBILL      2    26      1823  70.11827   0.9233   0.9204


                         MODEL Procedure
                         SUR Estimation

                  Nonlinear SUR Parameter Estimates

                                Approx.       'T'   Approx.
     Parameter     Estimate     Std Err     Ratio  Prob>|T|

     GA            3.192315     0.17650     18.09    0.0001
     GB            9.096972     2.22971      4.08    0.0004
     PA            5.220201     0.72099      7.24    0.0001
     PB           30.544008     1.28434     23.78    0.0001
     PC           -0.215128     0.01453    -14.81    0.0001
```

The Estimate Statement

An ESTIMATE statement has been added to the MODEL procedure to compute estimates and statistical properties of expressions involving parameters. For example the heat loss equation 6 has the parameter *R* but in the gas bill model you estimated GA, which is inversely proportional to R. If you want an estimate of the average R value of your house from your model, you need the following constants

$$\begin{aligned} hours\ per\ month &= 744 && (10)\\ cost\ per\ btu &= 6.348e-6 && (11)\\ area\ of\ house &= 8000 && (12) \end{aligned}$$

So R can be computed by

$$\begin{aligned} GA &= hours\ per\ month * cost\ per\ btu *\\ &\quad area\ of\ house * 1/R && (13)\\ &= 37.783 * 1/R && (14)\\ R &= 11.84 && (15) \end{aligned}$$

With these constants, you can use the following ESTIMATE statement in the MODEL procedure to estimate R and its standard errors:

```
estimate "R" 37.783 /ga;
```

The output is shown in Output 8.

Output 8. ESTIMATE Statement Output

```
                               The SAS System

                               MODEL Procedure
                               SUR Estimation

                          Nonlinear SUR Estimates

                                Approx.       'T'    Approx.
Term       Estimate      Std Err      Ratio   Prob>|T|  Label

R        11.835613      0.65438      18.09     0.0001  37.783 / GA
```

Tests on Parameters

When you perform an estimation, you are usually concerned about the statistical properties of the estimated parameters. All the procedures surveyed in this paper, with the exception of the NLP procedure, report standard errors of the estimates, and the T-ratio and the significance level for the test that the estimates are equal to zero. The significance level is based on a Wald test. For linear models, the statistical properties of the estimated parameters are explained by the exact test distribution. For nonlinear models, the statistical properties of the estimated parameters are only asymptotically valid.

A TEST statement has been added to the MODEL procedure to perform hypothesis tests on combinations of parameters using the Wald, lagrange multiplier, and likelihood ratio test methods. Performing the lagrange multiplier and likelihood ratio test methods using the other nonlinear procedures requires the ability to enforce parameter restrictions. Parameter restrictions are discussed in the next section.

Continuing from the example in the previous section, to test the null hypothesis that R = 10.5, you use the following statement:

```
test 37.783/ga = 10.5;
```

The output is shown in Output 9.

Output 9. TEST Statement Output

```
                          The SAS System

                          MODEL Procedure
                          SUR Estimation

                           Test Results

        Test                         Wald       Chi Prob.

        37.783 /GA = 10.5           4.166          0.041
```

Bounds and Restrictions on Parameters

As mentioned previously parameter restrictions are useful for testing hypotheses. Parameter restrictions are also used to enforce other constraints on a problem such as the sum of model parameters equaling one.

Some nonlinear estimation problems can have multiple solutions. Some of these solutions may be unrealistic. Parameter bounds are used to force an estimation to a particular solution.

For example, consider the following pharmacokinetic model used to determine the rates that Tetracycline, given orally, enters and leaves the blood stream:

$$\begin{aligned} conc &= dose * ka * (\text{exp}(-ka * t) - \text{exp}(-ke * t)) \\ &\quad /(-ka + ke) \qquad (16) \end{aligned}$$

The rate constants KA and KE must be positive. Using the NLP procedure, you can enforce this requirement with the BOUNDS statement. The following SAS statements estimate KA and KE:

```
proc nlp data=tetra;
   parms  ka ke;
   bounds ka > 0, ke > 0;
   lsq z;

   dose = 10;
   z = dose * ka * ( exp(-ka*t) - exp(-ke*t)) /
        (-ka+ke) - conc;

run;
```

Convergence

Convergence does not refer to an event signified by the alignment of the planets, and it is generally not as rare. For nonlinear estimation, there is generally no explicit formula for the estimates. The estimates must be determined iteratively. If nothing goes wrong, the iterative scheme you use will stop when it has found a solution (converged).

In the following sections, causes and solutions of convergence problems are discussed.

Minimization algorithms

You remember from calculus that the minimum (or maximum) of a function of one variable can be found by looking for the zeros of the first derivative of that function. The minimization algorithms used for nonlinear estimation do just that, they search for the zeros of the gradient of the objective function. The gradient is a vector of first derivatives of the objective function. A matrix called the *Hessian* (H), which represents how a small change in a parameter value changes the gradient, is used by most of the minimization algorithms to determine how to change the parameter values to reduce the gradient.

The Marquardt, Gauss-Newton, and Newton-Raphson methods are the most common algorithms used by the surveyed procedures. Other algorithms available (mainly in the NLP procedure) are

- Nelder-Mead Simplex
- Quadratic
- Quasi Newton
- Conjugate Gradient
- Double Dogleg
- Trust Region
- Hybrid Quasi-Newton (Least Squares)

Collinearity

Collinearity is not something reserved for linear problems. In fact, even though the final estimates for a nonlinear model may not be collinear, some sets of initial parameters to the model may be. The reverse is true as well.

Collinearity at its best slows down the convergence of the model. At its worst, it yields *biased* results.

Inadequate convergence criteria

There are many nonlinear functions for which the objective function is quite flat in a large region around the minimum point, so many different parameter vectors may satisfy a weak convergence criterion. By using different starting values, different convergence criteria, or different minimization methods, you can produce very different estimates for such models.

The GENMOD and NLIN procedures use the change in the parameter values as a convergence criterion. When the relative change in the parameter values is small, convergence is assumed. The NLIN procedure uses the change in the objective value (usually SSE) as a convergence measure by default. Both of these convergence measures are prone to signify convergence when the iteration process is slow and a solution has not yet been obtained.

A more effective convergence criterion is to stop the iterations when the Hessian provides no more information on how to change the parameters to reduce the gradient. This value is normalized so that the convergence measure is not fooled by the scale of the problem. This convergence measure is the primary convergence measure in the MODEL procedure and is one of the many convergence criteria used by the NLP procedure. In the NLP procedure the convergence measure is called GCONV or GTOL, and in the MODEL procedure it is called R.

Grid Search

Unlike linear estimations, nonlinear estimation can have multiple solutions. The solution a procedure finds could be a local minimum rather than a global minimum. You can attempt to avoid this situation by performing a grid search or by using carefully selected initial values.

The selection of starting values for an estimation can make the difference between convergence and nonconvergence. All the procedures surveyed offer some kind of *grid search* to aid in finding good starting values and in locating a global minimum. Grid search refers to using a set of initial guesses for parameters and determining which combination of those initial guesses produces the best starting point. The best starting point is determined by the smallest value of the objective function.

The MODEL procedure goes one step further and allows for minimization iterations to be performed for each combination of the initial guesses. The following is an example of the use of grid search in the MODEL procedure on the power bill model.

```
title1 'Power Usage Modeling';

/* First try - Simple initial guess ----- */
proc model data=gaspwr;
   parms pa pb pc ;
   control t_inside_summer 70;

   dt = avgtemp - t_inside_summer;
   if t_inside_summer > avgtemp then
       powbill = pb;
   else
       powbill = pa* dt* dt* exp(pc * dt)
                 + pb;

      /*---- Try again with grid search */
    fit powbill start=( pc 0 -1 -2 )/
         / itprint startiter=2;
run;
```

The parameter PC was given an initial value for the original estimation because with the default values the estimation failed to converge. In the new estimation, the value of PC is varied from 0 to -2 and two iterations for each guess are requested by the STARTITER=2 option.

Output 10 shows the output from the grid search. The output is requested by the ITPRINT option. Note that if no grid search iterations had been selected for the grid search, the selected initial values from the grid search would have resulted in a failed estimation.

Output 10. Grid Search with ITPRINT and STARTITER

```
                         The SAS System

                         MODEL Procedure
                         OLS Estimation

Estimates at Each START= Iteration

Iter     N Criterion Objective         PA         PB         PC
   0    28    0.9667  2285.046     0.0001     0.0001          0
   1    28         0  149.6008   0.473968  35.308931          0
   0    28    0.3773  521.3781   0.473968  35.308931  -1.000000
   1    28         0  447.1416  27.279486  40.416377  -1.000000
   0    28    0.0902  472.3415  27.279486  40.416377  -2.000000
   1    28 1.0537E-8  468.4969  35.300208  42.293147  -2.000000


                         The SAS System

                         MODEL Procedure
                         OLS Estimation

OLS Estimates at Each GAUSS Iteration

Iter     N Criterion Objective         PA         PB         PC
   0    28    0.7537  149.6008   0.473968  35.308931          0
   1    28    0.7928  123.3537   0.834698  34.528745 -0.0657828
   2    28    0.8150  108.0507   1.380115  33.739513  -0.118580
   3    28    0.8214  94.25683   2.695663  32.306431  -0.189092
   4    28    0.4993  40.35526   5.129560  30.478993  -0.228473
   5    28    0.0412  30.27973   5.406289  30.394732  -0.217061
   6    28  0.000359  30.22820   5.364843  30.409429  -0.217281
NOTE: At OLS Iteration 6 CONVERGE=0.001 Criteria Met.
```

Summary of Procedures Used for Nonlinear Regression

Table 1 provides a summary of the SAS procedures used for nonlinear regression. The **Rel.** column is the release in which the procedure became production. The newest procedure, NLP, was available experimentally in Release 6.08. The **Bounds** column indicates whether the procedure contains a BOUNDS statement. The MODEL procedure will have a bounds statement in an upcoming release. With the exception of the NLIN procedure, the bounds can be nonlinear. The **Systems** column indicates whether the procedure handles systems of equations.

Table 1. Summary of Procedures Used for Nonlinear Regression

Proc	Prod	Rel.	Bounds	Systems	Ders
GENMOD	STAT	6.09	no	no	analytic
MODEL	ETS	6.04	no	yes	analytic
NLIN	STAT	5.16	yes	no	no
NLP	OR	6.10	yes	yes	analytic numeric

The **Ders** column indicates what kind of derivatives are provided by the procedures. Derivatives of the residuals with respect to the parameters are necessary for the efficient estimation of nonlinear models. Derivatives are also necessary for solving for unknown variables in a nonlinear system and for nonlinear bounds, tests, and restrictions.

Summary of Nonlinear Estimation Methods

Table 2 summarizes the methods available for nonlinear estimation. The iterated version of the estimation methods (for example, ITSUR) is omitted from the table for conciseness.

Table 2. Summary of Provided Estimation Methods

Type	GENMOD	MODEL	NLIN	NLP
OLS	no	yes	yes	yes
SUR	no	yes	no	no
2SLS	no	yes	no	no
3SLS	no	yes	no	no
GMM	no	yes	no	no
ML	yes	yes	no	no
General optimizaton	no	no	no	yes

You noted previously that the NLP procedure is the most general nonlinear estimation procedure discussed, but Table 2 indicates that the NLP procedure does only OLS and general optimization. Many of the other estimation methods can be done in NLP, but you must explicitly write the objective function.

Nonlinear Solutions

In addition to the estimation of the model, you are sometimes concerned about the usefulness of the estimated model itself, either for forecasting or goal seeking.

For example, you could use the power/gas model to forecast how global warming would effect your utility bills. There are several problems with using the power/gas model to forecast. One problem is that the model does not fit the data perfectly nor is it expected to forecast correctly. Finally, the model does not contain the effect of inflation or the fact that the author got married recently. We can ignore the last two problems for now and consider them separately.

The errors in a forecast and the effects of random dependent variables (the average temperature) are simulated with the Monte Carlo capabilities of the MODEL procedure. The following example uses Monte Carlo simulation to generate error bounds on a forecast using the gas and power model.

```
proc model data=gaspwr outparms=est;
   parms ga gb pa pb pc -0.7;
   control t_inside_winter 65
           t_inside_summer 70 deltat=3;

   avgt = avgtemp + deltat * ranuni(546);

   if t_inside_winter < avgt then
      gasbill = gb;
   else
      gasbill = ga * ( t_inside_winter - avgt )
                + gb;

  dt = avgt - t_inside_summer;
  if t_inside_summer > avgt then
      powbill = pb;
   else
      powbill = pa* dt* dt* exp(pc * dt)  + pb;

  totcost = powbill + gasbill;

  fit powbill gasbill / sur outest=gasest outcov
              outs=s method=marquardt;
  id date;

  solve powbill gasbill / estdata=gasest sdata=s
        random=100 seed=123 out=monte ;
  outvars totcost;
run;
```

You have defined a new variable, ARGT, which is equal to AVGTEMP for the fit stage and is equal to AVGTEMP plus a random uniform error ranging from 0 to 3. The resulting Monte Carlo simulation will include errors that are due to fluctuations in average temperature and innate errors associated with the fit of the model itself.

The data for the plot in Figure 8 was generated by the following SAS statements:

```
proc sort data=monte;
   by date;
run;

proc univariate data=monte noprint;
   by date;
   var totcost;
   output out=bounds mean=mean p5=p5 p95=p95;
run;

data m; set monte;
   if _REP_ = 0 then output;
run;

data b; merge m (keep=totcost) bounds;
run;
```

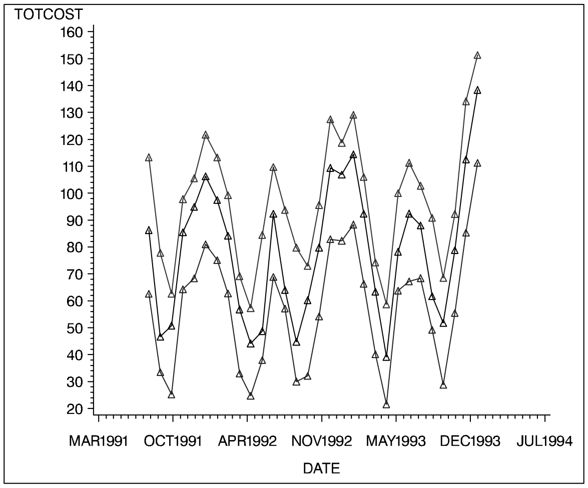

Figure 8. Prediction of TOTCOST with Error Bounds

Goal Seeking

You could also do goal seeking to find what the indoor temperature needs to be for the total monthly cost to be $80.50 when the temperature outside averages 80 degrees and $75.50 when the temperature outside averages 90 degrees. First, you set up a data set as in Figure 9.

```
data goal;
totcost =80.50; avgtemp=80;
output;
totcost =75.50; avgtemp=90;
output;
run;
```

Figure 9. Goal Seeking Data Set

The following SAS code is used is used to perform the goal seeking:

```
proc model parmsdata=est;
  parms ga gb pa pb pc ;
  control t_inside_winter 65 ;
  var t_inside_summer;

  if t_inside_winter < avgtemp then
      gasbill = gb;
   else
      gasbill = ga * ( t_inside_winter -
           avgtemp ) + gb;

  dt = avgtemp - t_inside_summer;
  if t_inside_summer > avgtemp then
      powbill = pb;
   else
      powbill = pa* dt* dt* exp(pc * dt)
                + pb;

  totcost = powbill + gasbill;

  solve t_inside_summer satisfy=totcost /
      data = goal solveprint;
run;
```

The output from the solution is shown in Output 11.

Output 11. Goal Seeking Output

```
Observation 1. NEWTON Iterations=10
   CC=5.258E-13(ERROR.TOTCOST=5.26E-13)

Solution Values:
T_INSIDE_SUMMER:   63.5051

Observation 2. NEWTON Iterations=5
   CC=4.277E-12(ERROR.TOTCOST=4.28E-12)

Solution Values:
T_INSIDE_SUMMER:   72.1799
```

Summary

This paper provided an overview of the SAS procedures that perform nonlinear estimation and simulation. SAS procedures discussed include MODEL, NLIN, NLP, and GENMOD.

References

Amemiya, T. (1985), *Advanced Econometrics*, Cambridge, MA: Harvard University Press.

Belsley, D.A.; Kuh, E.; and Welsch, R.E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons, Inc.

Chrzanowski, Charles, Private discussion February 1994.

Gallant, A.R. (1987), *Nonlinear Statistical Models*, New York: John Wiley and Sons, Inc.

Pindyck, R.S. and Rubinfeld, D.L. (1981), *Econometric Models and Economic Forecasts*, Second Edition, New York: McGraw-Hill Inc.

SAS Institute Inc. (1993), *SAS/ETS User's Guide, Version 6*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/STAT User's Guide: Release 6. version 4*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1993), *SAS Technical Report P-243, SAS/STAT Software: The GENMOD Procedure*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1993), *The NLP Procedure: Release 6.08 Extended User's Guide*, Cary, NC: SAS Institute Inc.