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:

\[ \text{conc} = \text{conc}_0 \exp(\text{rate} \times t) \]  

where \( \text{conc}_0 \) is the initial concentration, \( t \) is time, and \( \text{rate} \) 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.

\[ \log(\text{conc}) = \log(\text{conc}_0) + \text{rate} \times t \]  

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 \( \text{conc}_0 \).

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
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.

**Example of Time Series Estimation**

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

**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 $\epsilon_t$, the error of the model at time $t$, depends on $\epsilon_{t-i}$ or, more generally, the $\epsilon_t$’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 $\epsilon_t$’s are not identically distributed because the variance of $\epsilon$ 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.
National Weather Service, you can fit the following model using the MODEL procedure:

\[
\text{avgtemp}_t = a \times \text{avgtemp}_{t-12} + b \times \text{avgtemp}_{t-6} + c; \quad (3)
\]

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:

```sas
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

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 (\( \epsilon_i \)) are equal. Using the Breusch-Pagan test and an \( \alpha = 0.05 \), you cannot reject the null hypothesis.

```sas
weight _OBS_;
```

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

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 5. Residuals from OLS

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:

```sas
weight _OBS_;
```
Two-stage least squares, or 2SLS, is used to avoid simultaneity bias. For example, consider the biased and inconsistent parameter estimates. This is called least-squares estimation of these equations will produce model has dependent regressors, then nonlinear ordinary must be careful in choosing an estimation method. If the model has more than one dependent variable, you

The new set of estimates is shown in Output 6.

Output 6. GMM Time Series Regression Output

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:

\[ y_1 = a_1 + b_1 y_2 + c_1 x_1 + \epsilon_1 \]
\[ y_2 = a_2 + b_2 y_1 + c_2 x_2 + \epsilon_2 \]

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 \( \epsilon_1 \). Likewise, \( y_1 \) depends on \( \epsilon_2 \) 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 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

\[ \text{rate of heat flow} = \frac{A}{R} \cdot \Delta T \]  

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 \( \Delta T = (T_{\text{outside}} - T_{\text{inside}}) \).

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:

\[ \text{run time} = K (\Delta T)^s / \exp(s\Delta T) \]  

where \( K \) and \( s \) are parameters that depend on the properties of the refrigerant and compressor, and \( \Delta T \) 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:

\[ \text{gas bill} = ga \times (\text{rate of heat flow}) + gb \]  

(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:

\[ \text{pow bill} = pa \times (\text{runtime}) + pb; \]  

(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;
runc;
```

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

<table>
<thead>
<tr>
<th>Term</th>
<th>Estimate</th>
<th>Std Err</th>
<th>'T'</th>
<th>Approx. Prob</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>11.835613</td>
<td>0.65438</td>
<td>18.09</td>
<td>0.0001</td>
<td>37.783 / GA</td>
</tr>
</tbody>
</table>

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

\[ \begin{align*}
\text{hours per month} &= 744 \quad (10) \\
\text{cost per btu} &= 6.348 - 6 \quad (11) \\
\text{area of house} &= 8000 \quad (12)
\end{align*} \]

So \( R \) can be computed by

\[ GA = \frac{\text{hours per month} \times \text{cost per btu} \times \text{area of house}}{1/R} \]

(13)

\[ R = 11.84 \]

(15)

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

<table>
<thead>
<tr>
<th>Term</th>
<th>Estimate</th>
<th>Std Err</th>
<th>'T'</th>
<th>Approx. Prob</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>11.835613</td>
<td>0.65438</td>
<td>18.09</td>
<td>0.0001</td>
<td>37.783 / GA</td>
</tr>
</tbody>
</table>

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:

\[
\text{test } 37.783/\text{GA} = 10.5;
\]

The output is shown in Output 9.

**Output 9. TEST Statement Output**

<table>
<thead>
<tr>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
</tr>
<tr>
<td>37.783 /GA = 10.5</td>
</tr>
</tbody>
</table>

**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:

\[
\text{conc} = \frac{\text{dose} \times \text{ka} \times (\exp(-\text{ka} \times \text{t}) - \exp(-\text{ke} \times \text{t}))}{(-\text{ka} + \text{ke})} 
\]

(16)

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:

```sas
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.

```sas
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;
   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

<table>
<thead>
<tr>
<th>Iter</th>
<th>N</th>
<th>Criterion</th>
<th>Objective</th>
<th>PA</th>
<th>PB</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>28</td>
<td>0.9667</td>
<td>2285.046</td>
<td>0.0001</td>
<td>35.308931</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>0.3773</td>
<td>521.3781</td>
<td>0.473968</td>
<td>35.308931</td>
<td>-1.000000</td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>0.0902</td>
<td>472.3415</td>
<td>0.473968</td>
<td>40.416377</td>
<td>-1.000000</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>1.0537E-8</td>
<td>468.4969</td>
<td>35.300208</td>
<td>42.293147</td>
<td>-2.000000</td>
</tr>
</tbody>
</table>

The SAS System

MODEL Procedure
OLS Estimation

<table>
<thead>
<tr>
<th>Iter</th>
<th>N</th>
<th>Criterion</th>
<th>Objective</th>
<th>PA</th>
<th>PB</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>28</td>
<td>0.7537</td>
<td>149.6008</td>
<td>0.473968</td>
<td>35.308931</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>0.7926</td>
<td>123.3537</td>
<td>0.834698</td>
<td>34.528745</td>
<td>-0.065782</td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>0.8150</td>
<td>108.0507</td>
<td>1.380115</td>
<td>33.739513</td>
<td>-0.1185</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>0.8214</td>
<td>94.25683</td>
<td>2.695663</td>
<td>32.306431</td>
<td>-0.189092</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>0.4993</td>
<td>40.35526</td>
<td>5.129560</td>
<td>30.478993</td>
<td>-0.228473</td>
</tr>
<tr>
<td>5</td>
<td>28</td>
<td>0.0412</td>
<td>30.27973</td>
<td>5.406289</td>
<td>30.394732</td>
<td>-0.217061</td>
</tr>
<tr>
<td>6</td>
<td>28</td>
<td>0.000359</td>
<td>30.22820</td>
<td>5.364843</td>
<td>30.409429</td>
<td>-0.217281</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Proc</th>
<th>Prod</th>
<th>Rel.</th>
<th>Bounds</th>
<th>Systems</th>
<th>Ders</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENMOD</td>
<td>STAT</td>
<td>6.09</td>
<td>no</td>
<td>no</td>
<td>analytic</td>
</tr>
<tr>
<td>MODEL</td>
<td>ETS</td>
<td>6.04</td>
<td>no</td>
<td>yes</td>
<td>analytic</td>
</tr>
<tr>
<td>NLIN</td>
<td>STAT</td>
<td>5.16</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>NLP</td>
<td>OR</td>
<td>6.10</td>
<td>yes</td>
<td>yes</td>
<td>numeric</td>
</tr>
</tbody>
</table>

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.
You noted previously that the NLP procedure is the most
general nonlinear estimation procedure discussed, but Ta-
ble 2 indicates that the NLP procedure does only OLS and
general optimization. Many of the other estimation meth-
ods 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.

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:

```sas
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;
```

```sas
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;
```
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 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

<table>
<thead>
<tr>
<th>Observation 1.</th>
<th>NEWTON Iterations=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC=5.258E-13 (ERROR.TOTCOST=5.26E-13)</td>
<td></td>
</tr>
<tr>
<td>Solution Values:</td>
<td></td>
</tr>
<tr>
<td>T_INSIDE_SUMMER: 63.5051</td>
<td></td>
</tr>
<tr>
<td>Observation 2.</td>
<td>NEWTON Iterations=5</td>
</tr>
<tr>
<td>CC=4.277E-12 (ERROR.TOTCOST=4.28E-12)</td>
<td></td>
</tr>
<tr>
<td>Solution Values:</td>
<td></td>
</tr>
<tr>
<td>T_INSIDE_SUMMER: 72.1799</td>
<td></td>
</tr>
</tbody>
</table>

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.
Gallant, A.R. (1987), Nonlinear Statistical Models, New
York: John Wiley and Sons, Inc.
Pindyck, R.S. and Rubinfeld, D.L. (1981), Econometric
6, Cary, NC: SAS Institute Inc.
SAS/STAT Software: The GENMOD Procedure, Cary,
NC: SAS Institute Inc.
SAS Institute Inc. (1993), The NLP Procedure: Release