ARIMA and AUTOREG changes:
by David DeLong

ARIMA PROCEDURE CHANGES

The ARIMA procedure may now be executed interactively at the statement level. Additionally, some minor changes have been made to the estimation methods.

The option CONVERSE may be specified on the PROC ARIMA statement to request that each IDENTIFY, ESTIMATE or FORECAST statement be executed immediately. In this case the execution of the previous statement, terminating PROC or DATA statement or by the end of the job. If this option is used with the BY statement then the BY statement must immediately follow the PROC statement. When the termination of the procedure is signaled, all by groups remaining in the data set will be run through the same sequence of statements as the first by group.

The algorithm for computing the estimation has been modified in some cases. A Kalman filter algorithm (Pearson 1980, Morf, Siddhu and Kailath 1974) has been added to compute the so-called unconditional sum of squares or exact sum of squares (S) for high order moving average models. PROC ARIMA now chooses between a Kalman filter algorithm and a hand Cholesky algorithm (Ansley 1978) to minimize the cost of calculating with the Kalman filter being cheaper for ARIMA models with high order moving averages. Since the direct calculation of S is not excessively expensive the backforecasting algorithm for approximating S is no longer used. Consequently the BACKLIM parameter now has an effect only for negative values. If a negative number of backforecasts is requested, the same number of initial residuals are omitted from the sum of squares or likelihood function. Omitting values may be useful for suppressing transients in transfer function models.

The method of estimation is selected by a new parameter on the ESTIMATE statement. Specify METHOD=CLS for so-called conditional least squares estimates; and METHOD=ML (or the option ML) will generate maximum likelihood and conditional least squares estimates.

A STABLE option has been added to the ESTIMATE statement. Specifying STABLE requests that the autoregressive and moving average models be limited to the stationary and invertible regions respectively. If, during the iterative estimation, the new parameter estimates are outside the stable regions, the change in the parameters will halved up to 10 times until the new parameters are stable. If, after 10 halvings, the new estimates are still unstable, a warning message will be printed, convergence will be assumed and the old parameter estimates used.

The forecasts from the FORECAST statement are now generated using the same computational method as was used for the estimation of the model. Previously forecasts were generated using a fixed number of backforecasts regardless of which estimation method had been chosen. Thus if METHOD=CLS is used to estimate the model, the forecasts will be conditional forecasts, otherwise the forecasts will be finite sample predictions (Harvey 1981, Newton 1981, Newton and Pagno 1982) assuming that the model parameters are known. For long time series the finite sample forecasts will be very similar to the conditional forecasts. However for short series the forecasts and standard error estimates can be different.

SAS/ETS is a trademark of SAS Institute Inc., Cary, N.C. 27511.
The ULS and ML estimates employ a Gauss-Marquardt algorithm to, respectively, minimize the sum of squares and maximize the log likelihood. The relevant optimization is done simultaneously for both the \( \beta \) and \( \phi \) parameters. The OLS estimates of \( \beta \) and the Yule-Walker estimates of \( \phi \) are used as starting values for these methods.

The Yule-Walker equations, solved to obtain estimates of \( \phi \) and a preliminary estimate of \( \sigma^2 \), are:

\[
\text{Rs} = -r
\]

where \( R \) is the Toeplitz matrix whose \( i, j \)th element, \( R_{i,j} \), is given by \( \tau_{i-j} \), the lag \( |i-j| \) sample autocorrelation, \( r \) is the vector \( \{\tau_1, \ldots, \tau_p\} \) and \( a \) is the preliminary estimate of \( \phi \). If a subset of lags is specified for the autoregressive model then only those rows and columns corresponding to the lags are used in the above system.

The sample autocovariances are computed from the noise values, \( n = y - X \hat{\beta} \), as the sum of all available lagged products of order \( j \) divided by \( m + j \) where \( m \) is the number of such products. If there are no missing values then \( m + j = T \), the number of observations. In this case the matrix of autocorrelations, \( R \), is at least positive semidefinite. If there are missing values then these estimates of \( r \) can yield an \( R \) matrix not positive semidefinite. If such estimates occur a warning message is printed and the estimates are tapered by exponentially declining weights until \( R \) is positive definite.

In all of the estimation methods the original data are transformed by the inverse of the Cholesky root of \( \Sigma \). Let \( L \) denote the Cholesky root of \( \Sigma \), then \( \Sigma^{1/2} \) is the lower triangular matrix with the \( p \) lower diagonal elements and \( L \) is positive definite. For an AR(\( p \)) model \( L^{-1} \) is a band diagonal matrix with \( p + 1 \) 'non-diagonal' rows at the beginning and the autoregressive parameters along the remaining rows. Thus if there are no missing values, after the first \( p - 1 \) observations the data are transformed as:

\[
z_t = a_1 z_{t-1} + a_2 z_{t-2} + \ldots + a_p z_{t-p}.
\]

If there are missing values then the submatrix of \( L \) consisting of the rows and columns with nonmissing values must be used to generate the transformations. In AUTOREG the transformation is done using a Kalman filter and the Cholesky root is never directly computed. The Kalman filter algorithm as it applies here is described in Harvey and Phillips [1984] and Jones [1982]. However for ease of presentation the remaining discussion is in terms of \( L \).

Let the transformed residuals be denoted as, \( \epsilon = n - L \epsilon \). Then the ULS estimates are computed by minimizing \( \epsilon' \epsilon \) and the ML estimates are computed by minimizing \( \epsilon' [L \Sigma^{-1} L'] \epsilon \). To use the Gauss-Marquardt algorithm derivatives of \( c \) or \( [L \Sigma^{-1} L'] \epsilon \) with respect to the parameters are needed. The derivatives with respect to \( \beta \) are \( [L \Sigma^{-1} L'] \) and \( [L \Sigma^{-1} L'] \) \( X' \) respectively and are computed by the same transformation as before. The derivatives with respect to \( \phi \) may be computed by using the recurrences obtained by differentiating the Kalman filter recurrences and the equations for the initial conditions.

The final estimates of the variance matrix for the parameter estimates are computed as \( \hat{\sigma}^2 = [J' J]^{-1} \) where \( J \) is the matrix of derivatives of \( \hat{\beta} \) with respect to the parameters for the ULS estimates or the matrix of derivatives of \( [J' J]^{-1} \) divided by \( [J' J]^{-1} \) for the ML estimates. \( \hat{\sigma}^2 \) is the estimate of the mean square error, \( S(T - j) \), where \( S = n \) is the rank of the \( J \) matrix. If \( J \) is of full rank then \( \hat{\sigma}^2 \) is the number of free parameters in the model not counting \( \sigma^2 \). Also computed in these two cases is \( s^2 = [X' L^{-1} X]^{-1} \) as an estimate of the variance matrix of \( \phi \) assuming that \( \phi \) is known. Also the statistic, AIC and SBC are computed as:

\[
\text{AIC} = T \log(S/T) + \log(2\pi) + 2f
\]

\[
\text{SBC} = T \log(S/T) + \log(2\pi) + f \log(T).
\]

AIC is Akaike's information criterion and SBC is Schwartz's information criterion. See Judge et al. [1980] and Schwartz [1978] for additional detail.

**Predicted Values**

AUTOREG can now produce two kinds of predicted values and corresponding residuals and confidence limits. The residuals in both cases are computed as the actual value minus the predicted value.

The first type of predicted values is obtained from only the structural part of the model as \( X \hat{\beta} \). These are useful in predicting values of new response time series which are assumed to described by the same model as the current response time series. The other predicted values use both the structural part of the model and the predicted values of the error process. These are useful in predicting future values of the current response time series. The alpha level for either set of confidence limits may be specified as a parameter on the OUTPUT statement.

The predicted values and confidence limits for the structural part of the model, \( \hat{z}_t \), are computed as

\[
\hat{z}_t = X \hat{\beta}
\]

and the upper and lower confidence limits as

\[
\hat{z}_t + t_{\alpha/2} \hat{\sigma} \quad \text{and} \quad \hat{z}_t - t_{\alpha/2} \hat{\sigma}
\]

where \( 2 \) is the vector of independent variables for the \( \beta \)th observation, \( \hat{\sigma}^2 \) an estimate of the variance of \( \hat{z}_t \) and \( \hat{z}_t \) is the upper \( \alpha \) percentage point of the \( t \) distribution with \( T-p \) degrees of freedom.

The value of \( \alpha \) is calculated as

\[
\alpha = \frac{2}{\hat{\sigma}^2 + \hat{\sigma}^2}
\]

if the estimation method was YW. Otherwise it is calculated as

\[
\alpha = \frac{2}{\hat{\sigma}^2 + \hat{\sigma}^2}
\]

where \( W \) is the \( k \times k \) submatrix of \( (J' J)^{-1} \) which corresponds to the regression parameters.

The predicted values and confidence limits for future observations, \( \hat{y}_t = X \hat{\beta} + \hat{\varepsilon}_t \), conditional on past values are computed as

\[
\hat{y}_t = X \hat{\beta} + \hat{\varepsilon}_t
\]

where \( \hat{\varepsilon}_t \) is the finite sample predictor of \( \varepsilon_t \) (Harvey 1981, Newton 1981, Newton and Pagno 1982) given the available past values of \( \hat{y}_t \), \( \hat{y}_{t-j} = 1, 2, \ldots, T-1 \) and the autoregressive model for \( \varepsilon_t \). If the \( p \) previous values of \( \hat{\varepsilon}_t \) are available then

\[
\hat{\varepsilon}_t = a_1 \hat{\varepsilon}_{t-1} + \ldots + a_p \hat{\varepsilon}_{t-p}
\]

where \( a_1, a_2, \ldots, a_p \) are the estimated \( \phi \) parameters. The upper and lower confidence limits are computed as

\[
\hat{y}_t + t_{\alpha/2} \hat{\sigma}
\]

and

\[
\hat{y}_t - t_{\alpha/2} \hat{\sigma}
\]

where \( w \) is computed as

\[
w^2 = \hat{\sigma}^2 + \hat{\sigma}^2
\]

where the value \( \hat{\sigma}^2 \) is the estimate of the variance of \( \hat{\varepsilon}_t \). If \( p \) past values of \( \hat{\varepsilon}_t \) are available then \( w \) will be one. At the start of the series and after missing values \( w \) will generally be greater than one.
The formulas above do not account for the fact that the autoregressive parameters are estimated except to adjust the degrees of freedom for the error sum of squares. In particular the confidence limits are likely to be somewhat too narrow. In large samples this is not probably an important effect but it may be appreciable in small samples. See Harvey (1981) for some discussion of this problem for AR(1) models.

REFERENCES


THE SYSLIN AND PDLREG PROCEDURES:

by Leigh Ilenen

The SYSLIN PROCEDURE

The SYSLIN procedure has been renamed SYSLINSYSTEM of LNEAR equations. The basic structure of the procedure has been changed and three estimation methods have been added. Estimates can now be produced using k-class, minimum expected loss (MEL), or iterated three stage least squares, in addition to ordinary least squares, two stage least squares, limited information maximum likelihood, three stage least squares, and seemingly unrelated or joint generalized least squares.

The k-class estimator for a normalized structural equation has the form

\[
\begin{bmatrix}
\beta \\
\gamma
\end{bmatrix} = \left[ X'X \right]^{-1} X'y
\]

where the structural equation is

\[
y = Y\gamma + X_1\beta + \epsilon
\]

and \(X'X\) is the projection on to the null space of the X matrix, \(X_y\) being the instruments. \(Y\) is the matrix of endogenous variables and \(X_1\) is the matrix of exogenous variables in the structural equation.

The MELO estimator is described in an article by Zellner and Park in the March 1970 issue of JASA. The estimator is a bayesian estimator and a member of the k-class estimators. Parameters are estimated one structural equation at a time. The estimates minimize a quadratic loss function involving the reduced form parameters for each equation. The value of \(k\) for an equation is

\[
1 + \frac{r}{n-r-m-2}
\]

where

\(r\) is the rank of the exogenous matrix
\(n\) is the number of observations
\(m\) is the number of endogenous variables in the equation minus 1

The estimates are consistent and have a limiting normal distribution. The risk is finite relative to the loss function. The MELO estimator is a weighted combination of OLS and 2SLS estimates and tends to 2SLS estimates as \(m\) tends to infinity.

In addition to the new estimation methods, several new features have been implemented. Reduced form coefficients will be printed if the REDUCED option is specified on the PROC SYSLIN statement. The MELO estimator is a weighted combination of OLS and 2SLS estimates and tends to 2SLS estimates as \(m\) tends to infinity.

The SYSLIN procedure can now be produced using k-class, minimum expected loss (MEL), or iterated three stage least squares, in addition to ordinary least squares, two stage least squares, limited information maximum likelihood, three stage least squares, and seemingly unrelated or joint generalized least squares.

THE PDLREG PROCEDURE

A new procedure in the Econometrics and Time Series Library in PDLREG (Polynomial Distributed Lag REGression). The procedure
allows lagged time effects to be fit in a regression using the Almon distributed lag model.

The Almon distributed lag model has the form

\[ Y_t = a + \sum_{i=0}^{L} \beta_i X_{t-i} \]

where restrictions are placed on the X coefficients as

\[ \beta_i = \sum_{k=0}^{D} a_k r^k \text{ for some } D \leq L. \]

These restrictions can be justified on the basis that an unknown function can be approximated arbitrarily well on a closed interval by a polynomial of sufficiently high degree.

The procedure allows multiple lagged variable effects to be fit. A lagged variable effect is specified as

\[ \text{Variable}(L, D, Q, P) \]

where

- \( L \) is the number of lag periods
- \( D \) is the maximum degree polynomial to be fit.
- \( Q \) is the minimum degree polynomial to be fit.
- \( P \) impose endpoint restrictions on the Almon distributed lag.

For example, \( Y = X(8,4,3,0) \) fits the model

\[ Y = a + \sum_{i=0}^{8} \beta_i X_{t-i} \]

subject to

\[ \beta_i = \sum_{k=0}^{4} a_k r^k \]

The possible endpoint restrictions are:

- if not specified, no end point restriction
- \( P = 1 \), \( \beta_{-1} = 0 \)
- \( P = 2 \), \( \beta_{-1} = 0 \) and \( \beta_{+1} = 0 \)

The possible endpoint restrictions are:

- if not specified, no end point restriction
- \( P = 1 \), \( \beta_{-1} = 0 \)
- \( P = 2 \), \( \beta_{-1} = 0 \) and \( \beta_{+1} = 0 \)

The PDLREG procedure is invoked and controlled by the following statements:

- PROC PDLREG options;
- MODEL dependent=effects/options;
- BY variables;
- OUTPUT OUT= P= RESTRICT restrictions;
- LAGLIST lags;

The restrict statement allows non-lagged parameters to be restricted. A parameter on the MODEL statement is NLAG= number. This parameter specifies the order of an autoregressive model. If the model contains only one dependent variable and missing values are only at the beginning and end of the data set, Yule-Walker estimation is used to find estimates of the autoregressive parameters. The data is then transformed and the model parameters re-estimated. The LAGLIST statement allows the user to restrict autoregressive parameters to zero. All lags that do not appear in the statement are constrained to be zero. The output statement allows residuals and predicted values to be output to a SAS dataset.

The SAS/ETS Nonlinear Modeling Procedures: Changes and Enhancements

By Mark Little, SAS Institute

My name is Mark Little and I am responsible for the nonlinear systems modeling procedures in SAS/ETS. The nonlinear systems modeling procedures SYSLIN, SIMLIN, and MODEL have been completely rewritten during the past year. A number of features have been added and several outstanding problems corrected.

Today I'll review some of the more important changes and additions. First, I'll discuss the changes and enhancements common to the three procedures, and also some additions to the programming statements allowed in model programs. Then I'll discuss some new features specific to PROC SYSLIN and to PROC SIMLIN.

General Changes

A major change with the new version is that PROC MODEL is no longer necessary. In previous releases of SAS/ETS, PROC SYSLIN and PROC SIMLIN have relied on PROC MODEL to parse and differentiate the model program which was later compiled into IBM machine code. Now, both SYSLIN and SIMLIN can compile the model program and take all necessary derivatives without need of PROC MODEL. Both SYSLIN and SIMLIN can also create a new output model file for use in later steps.

The SAS/ETS Nonlinear Modeling Procedures: Changes and Enhancements
For example, this logistic population growth model is estimated by PROC SYSLIN from data in the data set USPOP and then simulated by PROC SIMLIN with data from data set USPOP. The model is read and estimated by SYSLIN and then stored on the model file WORK.M by the OUTMODEL= option. The MODEL= option then causes SIMLIN to read the model from this file.

The model files processed by SYSLIN, SIMLIN, and MODEL= are different in the new version. These files are special SAS data sets that contain the compiled model program and parameter values. They can now be copied, moved, deleted, listed, and archived like other SAS data sets.

An important feature of the new version in the RANGE statement. Previously the period-of-fit for SYSLIN or the forecast period for SIMLIN was controlled by the input data set, and the input data set had to be subset carefully to get the range right. In the new version the RANGE statement controls the period-of-fit or forecast range. The syntax of the RANGE statement is:

```
RANGE variable = start-value TO stop-value
```

The RANGE statement automatically takes care of any needed lags so that the analysis starts at the right place.

The way that lags are initialized has also been improved. Now, when SYSLIN estimates the parameters for a subset of the equations of a model, only as many lags as are needed for the estimation are filled—no observations are lost unnecessarily in starting the lags. The LAGEXEC option is now obsolete.

Missing values are fully supported by the new version, and lags are now initialized to missing values instead of to zero. Since missing values are supported, the new version can also be more tolerant of computational errors in the model program. Computational errors now result in missing values and produce error messages that are more detailed than in previous releases.

Another change in the new version is that SYSLIN and SIMLIN now generate four special variables for each equation in the model. For example, an assignment to an endogenous variable Y will generate variables PRED.Y, RESID.Y, ERROR.Y, and ACTUAL.Y. PROC SYSLIN uses the RESID variables to compute the sum-of-squares objective function for the parameter estimation, and PROC SIMLIN uses the ERROR variables to compute the simultaneous solution of the equation system. These variables can also be used in the model program—for example, the RESID variables can be used to refer to moving average error terms.

```
PROC SYSLIN DATA=USPOP; RANGES=USPOP; EXOGENOUS Y: EXOGENOUS X; PRED A B M1 P2; RESID Y = A + B * X + M1 + ZLAG1(RESID Y) + M2 + ZLAG2(RESID Y); ACTUAL Y = A + B * X + M1 + M2; ERROR Y = RESID Y - ACTUAL Y; RANGES START=1500 VERBOSE=1 NESTED=1; RUN;
```

**Figure 2. An AGG Error Process.**

Moving average error processes are now fully supported without need for any special coding tricks. The moving average error is just the lag of the RESID variable. In this example, the MA(2) error process is added to a linear regression model. Because lags are now initialized to missing values instead of to zero, the LAG functions cannot be used for ARMA processes. Instead, the new lagging functions ZLAG and ZDIFF are used to replace the initial missing values with zeros. These ZLAG and ZDIFF functions work just like the LAG and DIF functions, but where LAG and DIF would return a missing value, ZLAG and ZDIFF return zero instead.

Compound names like RESID Y can also be used for parameter names and other variables in the model program. In multi-equation models they are convenient for constructing parameter names. You can easily keep track of which equation a parameter goes with by using the dependent variable name as a prefix. A LABEL statement can also be used to give descriptive labels in parameter names and variables. These labels are used on the printouts and for the output data sets.

Some new programming features have also been added. Most of the programming features of the SAS DATA step can now be used in model programs. The PUT statement is now available to help with debugging models. The STOP, DELETE, ABORT, LINK, RETURN, GOTO, and CALL statements have also been added.

Character variables and expressions are now allowed in the model program for documentation and debugging purposes. Character variables can also appear in IF and OUTVARS statements. Three new debugging options—FLOW, PRINT, and PRINTALL—are now available to trace the program execution. These options are turned on and off by the CALL SETUP statement.

**New PROC SYSLIN Features.**

Now for some new features of PROC SYSLIN. A FIT statement has been added to provide three features. First, an initial grid search for starting parameter estimates can be requested with the START= option. Second, the FIT statement allows control of the equations included in the estimation. And third, the FIT statement allows selection of subsets of parameters to be estimated by equation name instead of by parameter list.

The INSTRUMENT statement has been extended to allow more flexibility in selecting the instruments for instrumental estimation methods like two-stage-least-squares. I don’t have time to discuss instruments in detail, but here is the new syntax.

Any variable can now be used in the instruments list, including parameter names. When a parameter is named as an instrument, SYSLIN uses the column of the Jacobian matrix of partial derivatives with respect to the parameter as the instrument. This feature is convenient in some circumstances and will be discussed more fully in the next edition of the User’s Guide.

The Marquardt-Levenberg method has been added to SYSLIN. METHOD=MARQUARDT can be specified to use Marquardt’s method to minimize the objective function instead of the default METHOD=GAUSS. Other optimization methods may be added in the future.

SYSLIN can now create an output data set containing the residual or predicted values from the estimation. With the new version it is no longer necessary to run PROC SIMLIN to get residuals for plots.

**New PROC SIMLIN Features.**

PROC SIMLIN also has some new features. A FORECAST option has been added to PROC SIMLIN to control the use of available actual values for endogenous variables when using the model to forecast.

A common problem in forecasting with a dynamic simultaneous model is that often the forecast must begin several periods in the past because of the need to start the dynamic lags. Recent historical data is available only with a delay, and some series are available sooner than others. This often means that there is no sharp division between past and future, and that just simulating the model for future periods will not produce a correct forecast.

In the new version of PROC SIMLIN, if you specify the FORECAST option then any prior information for an endogenous variable contained in the input data is used as the solution value for that variable. In other words, if you already have data for an endogenous variable for a period, then use it as the predicted value. This feature will be explained in more detail in the next edition of the User’s Guide.

Random number generator functions are now available for Monte Carlo simulation. For SYSLIN these functions always return zero, while for SIMLIN a RANDOM option has been added to cause the solution to be replaced 'n' times for each BY group, with different values returned from the random number functions.
In this example, a normal random error with a standard deviation determined by the CONTROL variable STD is added to the equation for Y. The RANDOM=9 option causes PROC SIMLIN to simulate the dynamic path of the predicted Y values nine times with different random errors. PROC SIMLIN produces combined statistics for all of the replications. The results are then plotted. The special variable _REP_is used by PROC PLOT to label the nine random simulations.

I know that these random number functions do not fully meet the need for Monte Carlo simulation facilities in SIMLIN. In the future I hope to be able to add more complete Monte Carlo features, but for now these random number functions may allow a partial solution for some users.

A DYNAMIC=n option has been added to PROC SIMLIN to allow simulation of n-period-ahead forecasting. In n-period-ahead forecasting, you periodically make forecasts a fixed number of periods into the future. For example, using a quarterly model, you might make forecasts for two years (8 quarters) into the future, and publish these forecasts four times a year (once each quarter). The SYSNLIN DYNAMIC=n option allows you to see how good your forecasts would have been on the average if you had been forecasting in this way in the past using your model with the actual exogenous values.

Other Features

That is all I have to say on new features for the procedures. But before I finish, I want to say a word on polynomial distributed lags. It has been suggested that SYSNLIN should support Almon polynomial distributed lag specifications. Of course SYSNLIN has always supported PDLs and most other distributed lags. It doesn't generate them automatically, and coding lag distributions can be a little messy, but if you know what a polynomial distributed lag is and know how to use the modeling language, you can specify PDLs without much trouble and SYSNLIN will estimate them just fine. But to make Almon lag models easier to use, a %PDL macro will be distributed with the next release that will automatically generate the needed model code for polynomial distributed lags.

There has not been time, and I doubt that you have the patience, for me to discuss the various minor changes and enhancements to the MODEL-SYSNLIN-SIMLIN system, and so in this talk I have just covered the highlights. In particular, the new version contains improvements in the printed output and better error diagnostics that I have not had time to discuss. Also, the documentation in the next edition of the SAS/ETS User's Guide is greatly enlarged and will give a much more detailed explanation of the procedures and how to use them.

Finally, I should mention a new SAS procedure that is not part of the ETS library. A major need in practical forecasting work is to be able to compare forecasts produced under different assumptions. A new procedure for making comparisons of SAS data sets, PROC COMPARE, will be added to the base product in the next release. PROC COMPARE will allow easy comparisons of forecasts.

In the coming year I expect to continue to work on further enhancements to the ETS nonlinear systems modeling procedures. This is a very large area of modeling and forecasting application and there is clearly much more that can be done. I hope that those of you who use these procedures will give me the benefit of your experience with the new version, and will let me know your criticisms and suggestions for improvements and new features.