

# ALTERNATIVE METHODS OF ESTIMATING PIECEWISE LINEAR AND HIGHER ORDER REGRESSION MODELS USING SAS® SOFTWARE

Lawrence Marsh, Manjula Maudgal and Jaishankar Raman  
University of Notre Dame, Notre Dame, Indiana 46556

## 1. Abstract

The purpose of this paper is to demonstrate how to use SAS® to estimate a piecewise linear or higher order spline regression model. The method of estimation depends upon whether the join point is known in advance or has to be estimated and whether a linear or a higher order model is used. When the location of the join point is not known but some reasonable initial guess can be made, the Gauss-Newton method may be used to search for the join point as suggested by Gallant and Fuller. For quadratic or higher order models, the Newton-Raphson method provides a useful alternative. These methods are easy to apply to piecewise regression models using SAS® as demonstrated in this paper.

## 2. Case where the location of the joint point, X\*, is known in advance.

First consider the problem of a two-segment piecewise linear regression model when the point where the two lines are to cross, X\*, is known in advance. These two segments may be thought of as being controlled by two dummy variables, D<sub>t1</sub> and D<sub>t2</sub>, where D<sub>t1</sub> = 1 in the first segment and D<sub>t1</sub> = 0 elsewhere. Similarly, D<sub>t2</sub> = 1 in the second segment and D<sub>t2</sub> = 0 elsewhere.

$$y_t = [a_1 + b_1 X_t] D_{t1} + [a_2 + b_2 X_t] D_{t2} + \epsilon_t \quad (1)$$

Fit by ordinary least squares regression subject to the constraint that the two functions touch at the point X\*:

$$a_1 + b_1 X^* = a_2 + b_2 X^* \quad (2)$$

At X\* this piecewise linear function is continuous but its first derivative is not. In other words, b<sub>1</sub> is not equal to b<sub>2</sub>. If b<sub>1</sub> is greater than zero a downturn may be said to occur at X\*. On the other hand, if b<sub>1</sub> is less than zero and b<sub>2</sub> is greater than zero an upturn occurs at X\*. The constraint giving us equation (2) may be re-expressed as:

$$a_2 = a_1 + b_1 X^* - b_2 X^*$$

This may be directly substituted into equation(1) as follows:

$$y_t = [a_1 + b_1 X_t] D_{t1} + [a_1 + b_1 X^* - b_2 X^* + b_2 X_t] D_{t2} + \epsilon_t$$

We now expand this expression and factor out each of the unknown coefficients and use the fact that D<sub>t1</sub> + D<sub>t2</sub> = 1 to obtain :

$$y_t = a_1 + b_1 [X_t D_{t1} + X^* D_{t2}] + b_2 [X_t - X^*] D_{t2} + \epsilon_t \quad (3)$$

Now define two new variables, Z<sub>t1</sub> and Z<sub>t2</sub> as:

$$Z_{t1} = X_t D_{t1} + X^* D_{t2} \quad \text{and} \quad Z_{t2} = (X_t - X^*) D_{t2}$$

to allow for ordinary least squares on the following model:

$$y_t = a_1 + b_1 Z_{t1} + b_2 Z_{t2} + \epsilon_t \quad (4)$$

Using the resulting coefficient estimates,  $\hat{a}_1$ ,  $\hat{b}_1$  and,  $\hat{b}_2$ , the restricted estimate for a<sub>2</sub> is given by:

$$\hat{a}_2 = \hat{a}_1 + \hat{b}_1 X^* - \hat{b}_2 X^* \quad (5)$$

These are the restricted parameter estimates for the original model specified by equations (1) and (2).

## 3. Case where location of X\* is unknown but an initial guess is available.

Here we have the problem where X\* is unknown and must be treated as a parameter to be estimated. One way to estimate this model is to see this as a nonlinear model and estimate it using nonlinear least squares regression. The Gauss-Newton method provides a standard algorithm for estimating this nonlinear regression model.

In general consider the problem of estimating a restricted regression model expressed as a nonlinear equation such as one given in equation (3). This equation may be expressed as:

$$y_t = f(\beta)_t + \epsilon_t \quad (6)$$

where f(β)<sub>t</sub> is dependent on the X values and in this case β is the 4 x 1 column vector

$$\beta = [a_1, b_1, b_2, X^*]'$$

The function f(β)<sub>t</sub> may be expressed as a truncated Taylor series expansion around the initial value vector β<sub>(0)</sub> as follows:

$$f(\beta)_t \approx f(\beta_{(0)})_t + f'(\beta_{(0)})_t (\beta - \beta_{(0)})$$

where the second order and higher level terms have been discarded leaving this truncated Taylor series expression to be substituted into equation (6) as a first round linear approximation of the original nonlinear model given by equation (3).

The linearized model becomes:

$$y_t = f(\beta_{(0)})_t + f'(\beta_{(0)})_t (\beta - \beta_{(0)}) + \epsilon_t^* \quad (7)$$

Where the error term ε<sup>\*</sup> now includes the truncated Taylor series remainder term as well as the original error term. Equation (7) may be rearranged to isolate the unknown coefficient vector β on the right hand side as follows:

$$y_t - f(\beta_{(0)})_t + f'(\beta_{(0)})_t \beta_{(0)} = f'(\beta_{(0)})_t \beta + \epsilon_t^* \quad (8)$$

the left hand side of this expression may then be defined to

be  $y_{t(0)}$  so that:

$$y_{t(0)}^* = y_t - f(\beta_{(0)})_t + f'(\beta_{(0)})_t \beta_{(0)}$$

and equation (8) becomes:

$$y_{t(0)}^* = f'(\beta_{(0)})_t \beta + \varepsilon_t^* \quad (9)$$

This equation can now easily be estimated by the ordinary least squares estimation.

$$\beta_{(1)} \approx \{[f'(\beta_{(0)})_t]' [f'(\beta_{(0)})_t]\}^{-1} [f'(\beta_{(0)})_t]' y_{t(0)}^* \quad (10)$$

Of course, this is analogous to the usual ordinary least squares result:

$$\hat{\beta} = \{X'X\}^{-1} X'y \text{ where } X = [f'(\beta_{(0)})_t] \text{ and } y = y_{t(0)}^* .$$

Once the value  $\beta_{(1)}$  has been obtained, a new Taylor series expansion around  $\beta_{(1)}$  can be generated. This method can be repeated such that after the  $i$ th iteration we have:

$$\beta_{(i+1)} \approx \{[f'(\beta_{(i)})_t]' [f'(\beta_{(i)})_t]\}^{-1} [f'(\beta_{(i)})_t]' y_{t(i)}^* \quad (11)$$

These iterations may be continued until the Euclidean norm  $\|\beta_{(i+1)} - \beta_{(i)}\|$  divided by  $\|\beta_{(i)}\|$  is less than some small value designated as the convergence criterion. The final  $\beta_{(i+1)}$  vector is then reported as the nonlinear least squares estimates of  $\beta$ . This final set of model coefficient values

$\hat{\beta} = [\hat{a}_1, \hat{b}_1, \hat{b}_2, \hat{X}^*]'$  provide the estimated model coefficients including an estimate of the value of the join point  $X^*$ .

Alternatively, equation (7) may be rewritten as:

$$y_t - f(\beta_{(0)})_t = f'(\beta_{(0)})_t (\beta - \beta_{(0)}) + \varepsilon_t^* \quad (12)$$

$$\text{Now define } y_{t(0)}^{**} = y_t - f(\beta_{(0)})_t \quad (13)$$

such that:

$$y_{t(0)}^{**} = f'(\beta_{(0)})_t (\beta - \beta_{(0)}) + \varepsilon_t^* \quad (14)$$

Applying ordinary least squares to this model yields:

$$(\beta - \beta_{(0)}) \approx \{[f'(\beta_{(0)})_t]' [f'(\beta_{(0)})_t]\}^{-1} [f'(\beta_{(0)})_t]' y_{t(0)}^{**}$$

or, after the  $i$ th iteration:

$$\beta_{(i+1)} \approx \beta_{(i)} + \{[f'(\beta_{(i)})_t]' [f'(\beta_{(i)})_t]\}^{-1} [f'(\beta_{(i)})_t]' y_{t(i)}^{**} \quad (15)$$

this shows more clearly the adjustment being made to the  $\beta_{(i)}$  vector at each iteration. This result can also be obtained by substituting

$$y_{t(i)}^* = y_{t(0)}^{**} + f'(\beta_{(i)})_t \beta_{(i)} \quad (16)$$

into equation (11). To apply this Gauss-Newton method to the piecewise linear regression problem presented above, we first expand equation (3) as:

$$y_t = a_1 + b_1 X_t D_{11} + b_1 X^* D_{12} + b_2 X_t D_{12} - b_2 X D_{1+} + \varepsilon_t \quad (17)$$

The appropriate specification for  $f'(\beta_{(i)})_t$  for this problem is the  $1 \times 4$  row vector:

$$f'(\beta_{(i)})_t = \left[ \frac{\partial f(\beta_{(i)})_t}{\partial a_1} \quad \frac{\partial f(\beta_{(i)})_t}{\partial b_1} \quad \frac{\partial f(\beta_{(i)})_t}{\partial b_2} \quad \frac{\partial f(\beta_{(i)})_t}{\partial X^*} \right] \quad (18)$$

Applying this to equation (17) we get :

$$\frac{\partial f(\beta_{(i)})_t}{\partial a_1} = 1$$

$$\frac{\partial f(\beta_{(i)})_t}{\partial b_1} = X_t D_{1(i)} + X_{(i)}^* D_{2(i)}$$

$$\frac{\partial f(\beta_{(i)})_t}{\partial b_2} = (X_t - X_{(i)}^*) D_{2(i)}$$

$$\frac{\partial f(\beta_{(i)})_t}{\partial X^*} = (b_{1(i)} - b_{2(i)}) D_{2(i)}$$

Note that the  $i$  subscripts on the dummy variables  $D_{1(i)}$  and  $D_{2(i)}$  imply that these dummy variables must be re-defined at each iteration to take account of the new location,  $X_{(i)}^*$ , where the two linear regressions are joined. Thus,  $D_{1(i)} = 1$  if  $X_t < X^*$  and  $D_{1(i)} = 0$  otherwise. Similarly,  $D_{2(i)} = 1$  if  $X_t \geq X^*$  and  $D_{2(i)} = 0$  otherwise. Equation (15) is applied over and over again until the value of  $\beta_{(i+1)}$  is sufficiently close to  $\beta_{(i)}$  according to the convergence criteria. The final  $\beta_{(i+1)}$  vector generates the nonlinear least squares estimates of  $\beta$ . This final set of model coefficient values

$\hat{\beta} = [\hat{a}_1, \hat{b}_1, \hat{b}_2, \hat{X}^*]'$  provide the estimated model coefficients including an estimate of the value of the join point  $X^*$ .

PROC NLIN and the more recent PROC SYSNLIN do not allow IF statements to be used within the PROC to redefine the dummy variables at each iteration. Therefore, the nonlinear search for the spline fit must make use of standard regression procedures like PROC REG or PROC IML to carry out the needed analysis.

The following SAS computer programs demonstrate the use of the Gauss-Newton method using PROC REG:

#### SAS Program using PROC REGs OUTEST option:

```
DATA SPLINE1;
INFILE RAWDATA;
INPUT Y X;
XKNOT=5;          * INITIALIZE KNOT LOCATION;
D1=1; D2=0;      * CREATE DUMMY VARIABLES;
IF X GT XKNOT THEN DO;
D1=0; D2=1; END;
KEEP Y X D1 D2 XKNOT;
```

```
DATA SPLINE2; SET;          * CREATE;
DUM1=X*D1+XKNOT*D2;      * TRANSFORMED;
DUM2=(X-XKNOT)*D2;      * VARIABLES;
```

```
PROC REG OUTEST=COEFFS; * FIT SPLINE MODEL;
MODEL Y = DUM1 DUM2; *GIVEN FIXED KNOT;
```

```
DATA COEF; SET COEFFS;
A1=INTERCEP; B1=DUM1; B2=DUM2; KEEP A1 B1 B2;
DO I = 1 TO 23; OUTPUT; END;
```

```
DATA SETUP; MERGE SPLINE1 COEF;
YSTAR=Y+XKNOT*(B1-B2)*D2; *TRANSFORM;
XB1STAR=X*D1+XKNOT*D2; *VARIABLES FOR;
XB2STAR=(X-XKNOT)*D2; *GAUSS-NEWTON;
XKTSTAR=(B1-B2)*D2; *ESTIMATION;
KEEP YSTAR XB1STAR XB2STAR XKTSTAR;
```

MACRO ITERATE;

```
PROC REG OUTEST=BETAS;
MODEL YSTAR = XB1STAR XB2STAR XKTSTAR;
```

```
DATA BETA; SET BETAS;
A1=INTERCEP; B1=XB1STAR; B2=XB2STAR;
XKNOT=XKTSTAR;
KEEP A1 B1 B2 XKNOT;
DO I = 1 TO 23; OUTPUT; END;
```

```
DATA ALL; MERGE SPLINE1 BETA;
D1=1; D2=0; * REDEFINE;
IF X GT XKNOT THEN DO; * THE DUMMY;
D1=0; D2=1; * VARIABLES;
END;
YSTAR=Y+XKNOT*(B1-B2)*D2; * SET UP FOR;
XB1STAR=X*D1+XKNOT*D2; * NEXT ;
XB2STAR=(X-XKNOT)*D2; * REGRESSION;
XKTSTAR=(B1-B2)*D2;
%
```

```
ITERATE; ITERATE; ITERATE; ITERATE; ITERATE;
```

An example of using the above procedure is provided by regressing the percent women's unemployment against the capacity utilization rate for the U.S. economy. Capacity utilization ranges from about 75 percent to about 96 percent while women's unemployment ranges from about 3 percent to about 6.5 percent from 1948 through 1970. A potential knot location for a change in the regression slope occurs at about 88 percent capacity utilization. In particular the decline in women's unemployment as capacity utilization increases moderates somewhat at about the 88 percent capacity utilization level. A slower decline is observed after that point. The problem is to estimate more precisely the exact location of this change in the regression slope while at the same time estimating the coefficients of this piecewise-linear regression model. Using the ITERATE macro to estimate this model by the Gauss-Newton method resulted in convergence after just two iterations. The location of the knot was estimated to be at a level of capacity utilization of  $X^* = 87.54171832$ . The regression for women's unemployment when capacity utilization is less than this value for  $X^*$  is given by

$$y_t = 16.66032450 - 0.14062228 X_t + \epsilon_t$$

while that for women's unemployment above this level of capacity utilization is given by

$$y_t = 13.49184217 - 0.10442831 X_t + \epsilon_t$$

This provides a modest though simple example. Extensions to more complicated models is straightforward although

sometimes laborious.

The PROC REG procedure works well for carrying out the Gauss-Newton nonlinear search algorithm except for the problem of determining when the estimated coefficients have converged. Since the ITERATE macro cannot be put in a do loop, we cannot automatically rerun it and stop it once convergence is achieved. Instead it is necessary to invoke the ITERATE macro by repeatedly calling the macro by repeating the command over and over again.

An alternative approach is to use PROC IML to achieve the same result. Regression procedures can then be carried out in matrix form within a do loop. The following program demonstrates this approach:

**SAS Program using PROC IML matrix approach:**

```
DATA SPLINE1;
ONE=1;
INFILE RAWDATA;
INPUT Y X;
XKNOT=5; * INITIALIZE KNOT LOCATION;
D1=1; D2=0; * CREATE DUMMY VARIABLES;
IF X GT XKNOT THEN DO;
D1=0; D2=1; END; * CREATE;
DUM1=X*D1+XKNOT*D2; * TRANSFORMED;
DUM2=(X-XKNOT)*D2; * VARIABLES;
```

```
PROC IML;
USE SPLINE1;
READ ALL;
X=ONE || DUM1 || DUM2;
XTX = X' * X; * GIVEN XKNOT;
IXTX = INV(XTX); * RUN OLS TO;
BHAT = IXTX*X'*Y; * GET INITIAL;
A1=BHAT(1,1); * COEFFICIENT;
B1=BHAT(2,1); * ESTIMATES;
B2=BHAT(3,1);
X1KNOT=XKNOT*(1,1);
BHOLD=A1//B1//B2//X1KNOT;
```

```
START LOOP1;
DO I = 1 TO 50;
D1=J(23,1,1); * RE-CREATE;
D1=D1#(X1<XKNOT); * DUMMY;
D2=J(23,1,1); * VARIABLES;
D2=D2#(X1>=XKNOT); * EVERYTIME;
YSTAR=Y+XKNOT*(B1-B2)*D2;
XB1STAR=X1#D1+XKNOT#D2;
XB2STAR=(X1-XKNOT)*D2;
XKTSTAR=(B1-B2)*D2;
X=ONE || XB1STAR || XB2STAR || XKTSTAR;
XTX=X'*X; * RERUN OLS;
IXTX=GINV(XTX); * ON THE;
BFIT=IXTX*X'*YSTAR; * TRANSFORMED;
A1=BFIT(1,1); * VARIABLES;
B1=BFIT(2,1);
B2=BFIT(3,1); * PULL OUT NEW ESTIMATES;
X1KNOT=BFIT(4,1);
XKNOT=J(23,1,X1KNOT); * FORM NEW KNOT;
BHOLD=BHOLD || BFIT; * COLLECT RESULTS;
```

```
END;
FINISH;
RUN LOOP1;
```

```
BHOLD=BHOLD'; * PRINT OUT RESULTS;
CNames={ 'A1' 'B1' 'B2' 'XKNOT' };
PRINT BHOLD (I COLNAME = CNames I);
```

To check this alternative method the same example discussed above concerning women's unemployment and capacity utilization was used to demonstrate this program. The results were identical to those from the ITERATE method including convergence in just two iterations. To verify the stability of this convergence six hundred iterations were performed. The results after six hundred iterations of the do loop in PROC IML were identical to those generated in the second iteration.

#### 4. Estimating higher order piecewise regression models.

An alternative to methods that are designed to fit a model directly such as the Gauss-Newton method just described, are methods that are designed to minimize the sum of squared deviations or maximize the likelihood function such as the Newton-Raphson method. This section will use the Newton-Raphson method to minimize the sum of squared deviations in a manner appropriate for quadratic or higher order piecewise regression models. The Newton-Raphson method will not work for simple piecewise linear regression models because the required second derivatives are all zero for the linear model.

Using equation (6) define the sum of squared deviations as :

$$S(\beta)_t = \sum_{t=0}^n [y_t - f(\beta)_t]^2 \quad (19)$$

Again around a given initial value,  $\beta_0$ ,  $S(\beta)_t$  can be expressed as a truncated Taylor series expansion:

$$S(\beta)_t \approx S(\beta_0)_t + S'(\beta_0)_t (\beta - \beta_0)_t + (\beta - \beta_0)_t S''(\beta_0)_t (\beta - \beta_0)_t / 2!$$

Since we wish to minimize this sum of squared deviations, we take the first derivative of  $S(\beta)_t$  with respect to  $\beta$  and set it equal to zero:

$$\frac{\partial S(\beta)_t}{\partial \beta} \approx [S'(\beta_0)_t]' + S''(\beta_0)_t (\beta - \beta_0)_t = 0$$

Solving for  $\beta$  yields the expression:

$$\beta \approx \beta_0 - [S''(\beta_0)_t]^{-1} [S'(\beta_0)_t]' \quad (20)$$

To apply equation (18) take the appropriate derivatives of equation (17) as follows:

$$S'(\beta_0)_t = - 2 \sum_{t=0}^n [y_t - f(\beta_0)_t] f'(\beta_0)_t$$

$$S''(\beta_0)_t = 2 \left\{ \sum_{t=0}^n [f'(\beta_0)_t]' [f'(\beta_0)_t] - \sum_{t=0}^n [y_t - f(\beta_0)_t] f''(\beta_0)_t \right\}$$

These variables including  $y_t$ ,  $f(\beta_0)_t$ , and  $f'(\beta_0)_t$  are given above while  $f''(\beta_0)_t$  is defined as:

$$f''(\beta_0)_t =$$

$$\begin{bmatrix} \frac{\partial^2 f(\beta_0)_t}{\partial a_1^2} & \frac{\partial^2 f(\beta_0)_t}{\partial a_1 \partial b_1} & \frac{\partial^2 f(\beta_0)_t}{\partial a_1 \partial b_2} & \frac{\partial^2 f(\beta_0)_t}{\partial a_1 \partial X^*} \\ \frac{\partial^2 f(\beta_0)_t}{\partial b_1 \partial a_1} & \frac{\partial^2 f(\beta_0)_t}{\partial b_1^2} & \frac{\partial^2 f(\beta_0)_t}{\partial b_1 \partial b_2} & \frac{\partial^2 f(\beta_0)_t}{\partial b_1 \partial X^*} \\ \frac{\partial^2 f(\beta_0)_t}{\partial b_2 \partial a_1} & \frac{\partial^2 f(\beta_0)_t}{\partial b_2 \partial b_1} & \frac{\partial^2 f(\beta_0)_t}{\partial b_2^2} & \frac{\partial^2 f(\beta_0)_t}{\partial b_2 \partial X^*} \\ \frac{\partial^2 f(\beta_0)_t}{\partial X^* \partial a_1} & \frac{\partial^2 f(\beta_0)_t}{\partial X^* \partial b_1} & \frac{\partial^2 f(\beta_0)_t}{\partial X^* \partial b_2} & \frac{\partial^2 f(\beta_0)_t}{\partial X^* \partial X^*} \end{bmatrix}$$

Substituting in for these values provides the basis for jointly estimating both the regression coefficients and the location of the join point,  $X^*$ .

The following SAS computer program demonstrates the use of the Newton-Raphson method.

#### SAS Program Newton-Raphson matrix approach:

```

DATA SPLINE1;
ONE=1;
INFILE RAWDATA;
INPUT Y X;
XK=5; * INITIALIZE KNOT LOCATION;
D1=1; D2=0; * CREATE DUMMY VARIABLES;
IF X GT XK THEN DO;
D1=0; D2=1; END;
DUM=(2*(XK**3) - 3*(XK**2)*X+(X**3))*D2;
KEEP ONE Y X D1 D2 XK DUM;

PROC IML;
USE SPLINE1;
READ ALL;
XM=ONE || X || DUM;
XTX = XM' * XM; * GIVEN XKNOT;
IXTX = INV(XTX); * RUN OLS TO;
BHAT = IXTX*XM'*Y; * GET INITIAL;
A1=BHAT(1,1); * COEFFICIENT;
B1=BHAT(2,1); * ESTIMATES;
B2=BHAT(3,1);
X1=XK(1,1);
BHOLD=A1//B1//B2//X1;

START LOOP1;
DO I = 1 TO 250;
D1=J(23,1,1); * RE-CREATE;
D1=D1#(X<XK); * DUMMY;
D2=J(23,1,1); * VARIABLES;
D2=D2#(X>=XK); * EVERYTIME;
STAT=Y-A1-B1#X-B2#D2#
(2#X1##3-3#X#X1##2+X##3);
DERA1=TRACE(DIAG(-2#STAT));
DERB1=TRACE(DIAG(-2#X#STAT));
DERB2=TRACE(DIAG(-2#STAT#D2#
(2#X1##3-3#X#X1##2+X##3)));
DERX1=TRACE(DIAG(-2#STAT#B2#D2#
(6#X1##2-6#X#X1)));
F=DERA1//DERB1//DERB2//DERX1;
DERA1A1=TRACE(DIAG(2));
DERA1B1=TRACE(DIAG(2#X));

```

```

DERA1B2=TRACE(DIAG(2#D2#
(2#X1##3-3#X#X1##2+X##3)));
DERA1X1=TRACE(DIAG(2#B2#D2#
(6#X1##2-6#X#X1)));
DERB1A1=TRACE(DIAG(2#X));
DERB1B1=TRACE(DIAG(2#X##2));
DERB1B2=TRACE(DIAG(-2#X#D2#
(2#X1##3-3#X#X1##2+X##3)));
DERB1X1=TRACE(DIAG(2#X#B2#D2#
(6#X1##2-6#X#X1)));
DERB2A1=TRACE(DIAG(2#D2#
(2#X1##3-3#X#X1##2+X##3)));
DERB2B1=TRACE(DIAG(-2#X#D2#
(2#X1##3-3#X#X1##2+X##3)));
DERB2B2=TRACE(DIAG(2#(D2#
(2#X1##3-3#X#X1##2+X##3)##2)));
DERB2X1=TRACE(DIAG(-2#STAT#D2#(6#X1##2-
6#X#X1)+2#B2#D2#(2#X1##3-3#X#X1##2+X##3)
#(6#X1##2-6#X1#X)));
DERX1A1=TRACE(DIAG(2#B2#D2#
(6#X1##2-6#X#X1)));
DERX1B1=TRACE(DIAG(2#X#B2#D2#
(6#X1##2-6#X#X1)));
DERX1B2=TRACE(DIAG(-2#STAT#D2#(6#X1##2-
6#X#X1)+2#B2#D2#(2#X1##3-3#X#X1##2+X##3)
#(6#X1##2-6#X1#X)));
DERX1X1=TRACE(DIAG(-2#STAT#B2#D2#(12#X1-
6#X)+2#(B2#D2#(6#X1##2-6#X#X1)##2)));
H = (DERA1A1 || DERA1B1 || DERA1B2 || DERA1X1)
// (DERB1A1 || DERB1B1 || DERB1B2 || DERB1X1)
// (DERB2A1 || DERB2B1 || DERB2B2 || DERB2X1)
// (DERX1A1 || DERX1B1 || DERX1B2 || DERX1X1);
B = B - INV(H)*F;
A1=B(1,1);
B1=B(2,1);
B2=B(3,1); * PULL OUT NEW ESTIMATES;
X1=B(4,1);
XK=J(23,1,X1); * FORM NEW KNOT;
BHOLD=BHOLD || B; * COLLECT RESULTS;

END;
FINISH;
RUN LOOP1;

BHOLD=BHOLD'; * PRINT OUT RESULTS;
CNAMES={ 'A1' 'B1' 'B2' 'X1' };
PRINT BHOLD (1 COLNAME = CNAMES );

```

## 5. Conclusions

This paper suggests some alternative methods for estimating piecewise linear and higher order regression models. For this purpose restricted regression and nonlinear regression methods are demonstrated. The restricted least squares method was shown to be appropriate when the exact location of the joint point is known in advance. Otherwise, nonlinear estimation methods can be used to search for the location of the join point while simultaneously estimating the usual regression coefficients. The nonlinear estimation methods suggested make use of the Gauss-Newton and Newton-Raphson algorithms which are demonstrated using various SAS<sup>®</sup> procedures. The Gauss-Newton method may be used for both linear and higher order piecewise regressions while the Newton-Raphson method only works for quadratic or higher order piecewise regressions. Example programming using various SAS<sup>®</sup> procedures is given.

## 6. References

- Gallant, A.R. and Fuller, W.A. (1973), "Fitting segmented polynomial regression models whose join points have to be estimated," *Journal of the American Statistical Association*, 68, pp. 144-147.

The location of the knot was estimated to be at a level of capacity utilization of  $X^* = 84.3253$ . The regression for women's unemployment when capacity utilization is less than this value for  $X^*$  is given by

$$y_t = 16.0545 - 0.1332 X_t + \epsilon_t$$

while that for women's unemployment above this level of capacity utilization is given by

$$y_t = 20.7315 - 0.2164 X_t + 0.0000039 X_t^3 + \epsilon_t$$

These results are similar to the results obtained above using the ITERATE and PROC IML approaches for the strictly linear piecewise model.