Modified Quadratic Hill-Climbing with SAS /IML

By: Dennis Patrick Leyden

Leyden, Dennis P. "Modified Quadratic Hill-Climbing with SAS/IML," *Computer Science in Economics and Management (*now *Computational Economics)*, 1991, 4(1):15-31.

Abstract:
This article describes EZClimb, a set of SAS/IML steps useful in solving numerical optimization problems. The program uses the method of modified quadratic hill-climbing with either analytical or numerical derivatives to maximize a user-defined criterion function. Modified quadratic hill-climbing is one of the more powerful algorithms known for function optimization but is not widely available outside of the software package GQOPT. The efficacy of the SAS steps is illustrated using Rosenbrock's function, Savin and White's Box-Cox extended autoregressive model, and Klein's Model.
Key words. Hill-climbing, maximum likelihood estimation, non-linear least squares, optimization.
Abbreviations. BHHH: Berndt, Hall, Hall, and Hausman's optimization algorithm. FIML: full-information maximum-likelihood. SAS /IML: SAS Institute's Interactive Matrix Language. TSP: Time Series Processor software package.

Article:

1. Introduction

Empirical analysis increasingly involves the use of maximum-likelihood or non-linear least-squares estimation. Historically, nonlinear least squares has been used more because of the high costs associated with maximum likelihood (Belsley, 1979). However, recent econometric developments (for example, Belsley, 1974, 1979, 1980), in conjunction with new computer hardware and software have led to increasing use of maximum-likelihood methods. Today, both maximum-likelihood and non-linear least-squares estimation are commonly treated in econometric texts (Chow, 1983; Greene, 1990; Judge *et al.*, 1982; Kmenta, 1986; Maddala, 1977) and used in increasingly more sophisticated ways (for example, Moffitt, 1986).

One of the more powerful algorithms for tackling these problems is the method of modified quadratic hill-climbing (Goldfeld, Quandt, and Trotter, 1966, 1968). Unfortunately, this method is not widely available outside GQOPT, Goldfeld and Quandt's software package useful for a wide range of numerical optimization problems including several econometric applications. The purpose of this article is to describe a set of SAS steps in the form of a program called EZClimb which implements this algorithm in a manner accessible and useful to a broad range of researchers.[1] Use of this program is illustrated with three well known problems including the Rosenbrock function, an autoregressive Box-Cox likelihood function, and FIML estimation of Klein's Model I.

The issue of accessibility is not trivial. While there have been enormous strides in software development, most packages require a model to fit into one of several canonical forms. Where customization is possible, the documentation for that process is often cryptic. As a result, researchers often must compromise their model to fit the software. EZClimb provides a more accessible alternative which is applicable to a wide variety of problems including those not easily treated by current software packages.

The accessibility of EZClimb is due in large part to the use of SAS /IML code. Unlike other languages such as Fortran (often used in current software), SAS/ IML does not require such preliminaries as space allocation, dimensioning, or definition of variable type. Matrices are the basic unit with elements of matrices referred to in a standard row-column manner. Moreover, complex operations such as inversion, the calculation of eigenvalues, or the determination of row or column maxima can be performed in a straightforward way with a single command. As a result, SAS/IML code can be used easily by those familiar with matrix algebra.

This accessibility allows the program to handle a broad range of estimation problems. While it is applicable to most problems in which an objective function is maximized or minimized, it is particularly well suited for problems involving complex or idiosyncratic error structures such as those associated with likelihood functions in which the likelihood of particular observations takes one of several functional forms. Other examples include nonlinear least-squares problems and qualitative dependent-variable models such as probit, logit, and Tobit.

The program is sufficiently flexible to respond to the various needs its users. While default values are provided, the user has control over all parameters used in the estimation process. Derivatives can be calculated either analytically or numerically, and an optional statistical analysis based on asymptotic maximum-likelihood theory is also available. The structure of the program is such that the more sophisticated user can easily recode the program for alternative hill-climbing methods or an expanded statistical analysis. Indeed, the general accessibility of the program makes it a potentially useful tool in teaching hill-climbing estimation.

2. A Review of the Hill-Climbing Algorithm

EZClimb uses Goldfeld, Quandt, and Trotter's (1968) modified quadratic hill-climbing method as the foundation for its estimation procedure.[2] This method belongs to a class of estimation procedures called hill-climbing methods and which are based on the steepness and curvature of the function being maximized. Amemiya (1983), Goldfeld and Quandt (1972), Judge *et al.* (1985), Powell (1970), and Quandt (1983) provide reviews of this class of estimation techniques.

In hill-climbing, the goal is to maximize some criterion function *f*(p) by the appropriate choice of values for the parameter vector **p.** Let **F** denote the vector af/ap, S denote the matrix *32f/* ap2, and *t* denote the iteration. The method of quadratic hill-climbing is based on two observations: (1) the most direct way to improve the value of the function f(p) is to move uphill in the steepest direction (given by **F),** and (2) if the function were quadratic, the maximum could be found in one step (using **S**). The first observation leads to an iterative routine called the steepest-ascent method in which parameter values are changed by the first derivatives of the criterion function in order to increase the value of the function

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{F}_t \, . \qquad (1)$$

The second observation leads to an iterative routine often called the Newton—Raphson method in which parameter values are changed by the product of first and second derivatives

$$\mathbf{p}_{t+1} = \mathbf{p}_t - \mathbf{S}_t^{-1}\mathbf{F}_t \, . \qquad (2)$$

Neither method (1) nor (2) is ideal. The steepest-ascent method can be slow and may be unable to get past small valleys. Use of the Newton—Raphson method, on the other hand, while often able to overcome problems such as small valleys, will perform poorly if the criterion function is not closely approximated by a quadratic function and will fail if S is singular. The quadratic hill-climbing method combines these two methods by weighting more heavily the method which is more likely to succeed. When the criterion function is closely approximated by a quadratic function, the Newton—Raphson method is more heavily weighted. When the quadratic approximation is poor, the steepest-ascent method is weighted more heavily. Thus, the quadratic hill-climbing method is an iterative routine in which the change in parameter values is a combination of Equations (1) and (2) given by

$$\mathbf{p}_{t+1} = \mathbf{p}_t - (\mathbf{S}t - \alpha \boldsymbol{I})^{-1}\mathbf{F}_t \, . \qquad (3)$$

α is a negative function of the degree to which the criterion function can be approximated by a quadratic function.[3] When α equals zero, Equation (3) becomes identical to Equation (2); as α rises above zero, Equation (3) approaches Equation (1).

Two special cases may occur which cannot be handled by Equation (3). First, a may be negative. If this occurs, the quadratic approximation works well and the new parameter vector p_{t+1} is defined as

$$\mathbf{p}_{t+1} = \mathbf{p}_t - \mathbf{S}_t^{-1}\mathbf{F}_t \, . \tag{4}$$

Secondly, F_t may equal zero. If this occurs and S_t is negative definite, then a local maximum had been found, and the iterative process stops. However, if $F_t = 0$ and S_t is not negative definite, then the iterative process has found a saddlepoint or the bottom of a valley. In this case, both the steepest ascent and Newton—Raphson methods will fail. However, the step $\pm \lambda_1 U_1$ (where λ_1 is the largest eigenvalue of S_t and U_1 is the eigenvector associated with λ_1) maximizes the quadratic approximation of the criterion function over a sphere of radius λ_1 centered at $\mathbf{p_t}$**.** (See Lemma 3 and the Appendix to Goldfeld, Quandt, and Trotter (1966).) Thus, when $F_t = 0$ and S_t is not negative definite, the new parameter vector p_{t+1} is defined to be

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \lambda_1 U_1 \, . \tag{5}$$

Intuitively, Equation (5) allows the quadratic hill-climbing method to step away from a saddlepoint or valley floor in what is approximately the best direction so that it may continue its search process.

One criticism of the quadratic hill-climbing method is that it is likely to be slow in the presence of ridges because it searches for an improvement over a spherical region. The modified quadratic hill-climbing method allows the process to become quicker by changing the search region to an ellipsoidal one. This is accomplished by replacing the identity matrix I in Equation (3) with the positive definite matrix A_t so that $\mathbf{p}_{t+1}$ is defined to be

$$\mathbf{p}_{t+1} = \mathbf{p}_t - (\mathbf{S}_t - \alpha \mathbf{A}_t)^{-1}\mathbf{F}_t \, . \tag{6}$$

The construction of A_t is based on "the heuristically plausible assumption that the most useful direction of search at any point is close to the direction of the immediately preceding step.[4] Define δ_1 to be the preceding step

$$\delta t = \mathbf{p}_t - \mathbf{p}_{t-1} \tag{7}$$

and note that the future step δ_{t+1} can be decomposed into an orthogonal projection on the vector δ_t

$$\delta'_{t+1} = \frac{\delta_t^T \delta_{t+1}}{\delta_t^T \delta_t} \delta_t \tag{8}$$

and its orthogonal complement

$$\delta''_{t+1} = \delta_t - \delta'_{t+1} \, . \tag{9}$$

Using this decomposition, δ_{t+1} can be rewritten in a new coordinate system which emphasizes the direction δ_t by

$$\beta \delta'_{t+1} + \delta''_{t+1} = \mathbf{B}_t \delta_t \, , \tag{10}$$

which, given Equations (8) and (9), can be rewritten

$$\mathbf{B}_t \delta_t = \left[I + (\beta - 1) \frac{\delta_t \delta_t^T}{\delta_t^T \delta_t} \right] \delta_t \, . \tag{11}$$

We can, of course, search for an improvement in the criterion function using the new coordinate system defined by the B_t transformation. However, searching over a spherical region under the transformed coordinate system is equivalent to searching over an ellipsoidal region in the original coordinate system. Hence, using the B_t transformation to define an ellipsoidal search region and converting back to the original coordinate system, we find A_t is defined to be

$$\mathbf{A}_t = \mathbf{B}_t^T \mathbf{B}_t \, . \tag{12}$$

β determines the degree to which the direction δ_t is to be emphasized. When the quadratic approximation of the criterion function is poor, it is likely that the value of searching over an ellipsoidal region will be high. Hence, β is set relatively close to one. When the quadratic approximation of the criterion function is good, there is less need for an ellipsoidal search region and so the value of β is smaller. In essence, a greater value of β results in greater stretching of the ellipsoidal search region in the direction of δ_t.

In practice, there are several algorithm variables, such as β, whose initial values need to be set and whose later values are a function of prior successes. Two algorithm variables of particular interest with the modified quadratic hill-climbing algorithm are *R* (a scaling factor which is used in the calculation of α and which goes to zero as the quadratic approximation gets better) and *h* (a scalar initially set at one and then increased in order to speed up the estimation process by stretching the step $-(S_t - \alpha A_t)^{-1}F_t$. The actual implementation of Equation (6) thus becomes

$$\mathbf{p}_{t+1} = \mathbf{p}_t - h(\mathbf{S}_t - \alpha(R)\mathbf{A}_t)^{-1}\mathbf{F}_t \,. \qquad (13)$$

After having chosen initial values for the algorithm parameters, the user chooses a set of starting values, p_0. The user then iterates over Equation (13) (or Equations (4) or (5) if need be) until a maximum is found. A more formal statement of the algorithm is given in the Appendix.

A particular advantage of quadratic hill-climbing methods when engaged in maximum likelihood estimation is the use of S to estimate the information matrix. Although the information matrix is the *expectation* of the negative of the second derivative matrix, ignoring the expectation and using the negative of the second derivative matrix yields a consistent estimator of the information matrix (Cramer, 1986, pp. 27-28). This use of second derivatives to estimate the information matrix provides EZClimb with an advantage over programs like TSP which rely only on first-derivative methods in their maximum likelihood routines. The failure to use second derivatives in algorithms like BHHH, used for example by TSP in its FIML command, may generate inaccurate (though consistent) estimates of the maximum-likelihood standard errors. Spitzer (1984), for example, has shown that in finite samples the Hessian for the Box-Cox likelihood function cannot be correctly evaluated by the BHHH method and that, in fact, for Box-Cox likelihood functions, all first-derivative iterative estimation methods will overestimate variances (p. 651).[5] Thus, while programs like TSP may permit easy coding of a model for maximum-likelihood estimation, an algorithm like quadratic hill-climbing used in EZClimb is preferable when interest centers on accurate evaluation of the second-derivative matrix to obtain standard errors.

The EZClimb program provides an optional statistical analysis in which asymptotic standard errors are calculated based on the second-derivative matrix and the assumption that the Cramer-Rao lower bound is met.

3. Illustrative Examples

The use of the modified quadratic hill-climbing algorithm with EZClimb is fast, accurate, and straightforward. Consider three familiar problems drawn from the optimization and econometric literatures.

The first problem is the maximization of Rosenbrock's function. Long known as a test function (Rosenbrock, 1961; Fletcher and Powell, 1963; Goldfeld, Quandt, and Trotter, 1966, 1968; Goldfeld and Quandt, 1972), it was used with EZClimb to illustrate the general ability of the program to find a maximum, to test the efficacy of the numerical derivative modules, and to aid in the determination of the default values for the several variables used by the program.

The Rosenbrock function, which has a surface which resembles a narrow, U-shaped ridge, is defined by

$$z = -100(y - x^2)^2 - (1 - x)^2 \qquad (14)$$

and takes a maximum value of zero at the point (1, 1). Using either analytical or numerical derivatives, the program generally performed well in its search for the maximum. Table I provides a summary of the iterations. As can be seen, results using numerical derivatives closely approximated the results using analytical derivatives and were reasonably close to the results of Goldfeld, Quandt, and Trotter (1968).[6]

Table I. Evidence of numerical derivative accuracy – convergence pattern for two-dimensional Rosenbrock function

	EZClimb						Goldfeld, Quandt, and Trotter (1968)		
	Analytic derivatives			Numeric derivatives					
Iter.	x	y	z	x	y	z	x	y	z
0	−1.2000	1.0000	−24.200	−1.2000	1.0000	−24.200	−1.2000	1.0000	−24.200
1	−1.0317	1.0901	−4.1938	−1.0317	1.0901	−4.1938	−1.1071	1.1850	−4.6048
3	−0.7842	0.5688	−3.3963	−0.7842	0.5688	−3.3963	−0.5820	0.2964	−2.6819
5	−0.4060	0.1794	−1.9980	−0.4060	0.1794	−1.9981	0.2797	−0.0095	−1.2891
7	−0.0246	−0.0395	−1.2104	−0.2452	−0.0395	−1.2106	0.4272	0.1682	−0.3484
9	0.2629	0.0431	−0.6112	0.3427	0.0533	−0.8433	0.7005	0.4987	−0.0961
11	0.5435	0.2692	−0.2771	0.5213	0.2518	−0.2690	0.8402	0.7051	−0.0256
13	0.7384	0.5297	−0.0924	0.7550	0.5701	−0.0600	0.9490	0.9008	−0.0026
15	0.9111	0.8182	−0.0219	0.9823	0.9640	−0.0004	0.9934	0.9868	-10^{-5}
17	0.9923	0.9817	−0.0010	0.9984	0.9964	-10^{-5}	1.0000	1.0000	-10^{-13}
19	1.0000	1.0000	-10^{-8}	1.0000	1.0001	-10^{-9}	na	na	na
21	1.0000	1.0000	-10^{-29}	1.0000	1.0000	-10^{-20}	na	na	na

The second problem, taken from the econometric literature, was to estimate Savin and White's (1978) Box-Cox extended autoregressive model of consumption function derived from Klein's Model I. This problem was chosen to illustrate the EZClimb's ability to handle econometric problems with complicated likelihood functions. The likelihood function for this problem as given by Savin and White (1978, p. 5) is:

$$L(\lambda, \rho; y, \mathbf{X}) = -(21/2)(\ln(2\pi)+1) - (21/2)\ln(\hat{\sigma}^2(\lambda, \rho)) + (1/2)\ln(1-\rho^2) + (\lambda - 1)\sum_{t=1}^{21}\ln(y_t)\,, \qquad (15)$$

with $\hat{\sigma}^2(\lambda, p)$ defined in terms of the Box-Cox transformed data. The results of estimating this model using EZClimb along with Savin and White's results are reported in Table II. Derivatives were calculated numerically, and with one exception (the rate with which *h* is increased) only default values of the hill-climbing algorithm's variables were used. As can be seen, the results are identical to Savin and White's results without resorting to the two-dimensional grid search over λ and *p* that they employed. Note also that Savin and White's parameter estimates are only accurate to two places because of limitations inherent in the SHAZAM program.

Table II. Maximum likelihood estimation of the Box-Cox extended autoregressive model

	EZClimb[a]					Savin and White[b]
	1	2	3	4	5	
Starting values: λ	1.0000	−0.5100	1.0000	0.0000	−2.0000	na
ρ	0.0000	0.0000	0.4400	0.0000	0.0000	na
Estimates: λ	−0.48291	−0.48291	−0.48291	−0.48291	−0.48291	−0.48
ρ	0.22149	0.22149	0.22149	0.22149	−0.22149	0.22
$L(\lambda, \rho)$	−23.5019	−23.5019	−23.5019	−23.5019	−23.5019	−23.5019
E_{p_t}	0.04952	0.04952	0.04952	0.04952	0.04952	0.0495
$E_{p_{t-1}}$	0.01329	0.01329	0.01329	0.01329	0.01329	0.0133
E_w	0.62857	0.62857	0.62857	0.62857	0.62857	0.6286
Seconds of CPU time	87	87	101	129	158	na
Function Evaluations	192	199	233	291	383	na
Iterations	7	7	8	10	13	na

[a] Results generated on a VAX 8700 with the convergence criterion set at 7 (see Table V) and with tolerances set at 10^{-04}. Derivatives calculated numerically.
[b] Results generated using SHAZAM.

The last problem, which illustrates EZClimb's ability to handle simultaneous-equations models, was to estimate the well known Klein's Model I. Using the form estimated by Chow (1968) and Goldfeld and Quandt (1972) and using 21 observations in deviation-from-mean form, Klein's Model I can be written as:

$$U = YB + XA\,, \qquad (16)$$

where U is a 21 x 3 matrix of normally distributed errors, Y is a 21 x 3 matrix of dependent variables, X is a 21 x 7 matrix of independent variables, and B and A are the conformable coefficient matrices:

$$
\mathbf{B} = \begin{bmatrix} -1 & \beta_{21} & \beta_{31} \\ \beta_{12} & -1 & 0 \\ \beta_{13} & 0 & -1 \end{bmatrix} , \tag{17}
$$

$$
\mathbf{A} = \begin{bmatrix} \beta_{12} & 0 & 0 \\ \gamma_{12} & 0 & \gamma_{32} \\ -\beta_{13} & 0 & \gamma_{33} \\ 0 & \gamma_{24} & 0 \\ -\beta_{13} & \beta_{21} & 0 \\ \beta_{13} & 0 & 0 \\ 0 & \gamma_{27} & 0 \end{bmatrix} . \tag{18}
$$

The criterion function for this model is the likelihood function:

$$
\begin{aligned} &L(\beta_{12}, \beta_{13}, \gamma_{12}, \beta_{21}, \gamma_{24}, \gamma_{27}, \beta_{31}, \gamma_{32}, \beta_{33}; \mathbf{Y}, \mathbf{X}) \\ &\quad = -(1/2) \ln((1/21) \det(\mathbf{U}^T \mathbf{U})) + \ln(|\det(\mathbf{B})|) . \end{aligned} \tag{19}
$$

Table III. FIML estimation of Klein's model I. SV 1: (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0). SV 2: (0.20410, 0.10250, 0.22967, 0.72465, 0.23273, 0.28341, 0.23116, 0.54600, 0.85400).

Coefficient	EZClimb[a]		Chow	Goldfeld, Quandt[b] (1972)	
	SV 1	SV 2		SV 1	SV 2
β_{12}	−0.16079	−0.16079	−0.16079	−0.16426	−0.16957
β_{13}	0.81143	0.81143	0.81144	0.82053	0.81734
γ_{12}	0.31295	0.31295	0.31295	0.31561	0.30402
β_{21}	0.30568	0.30568	0.30569	0.31122	0.31977
γ_{24}	0.30662	0.30662	0.30662	0.30674	0.30357
γ_{27}	0.37170	0.37170	0.37170	0.37202	0.36948
β_{31}	−0.80101	−0.80101	−0.80099	−0.77638	−0.72073
γ_{32}	1.05185	1.05185	1.0519	1.05149	1.02007
γ_{33}	0.85190	0.85190	0.85190	0.85120	0.84839
Function value	−2.75551	−2.75551	na	−2.75562	−2.75625
Seconds of CPU time	150	98	na	42	24
Function Evaluations	7373	5258	na	3289	1830
Iterations	21	15	8	na	na
Tolerance	10^{-4}	10^{-4}	na	na	na

[a] Estimation performed on a VAX 8700 with the convergence criterion set at 7 (see Table V) and with tolerances set at 10^{-04}. Derivatives calculated numerically.
[b] Estimation performed on an IBM 7094.

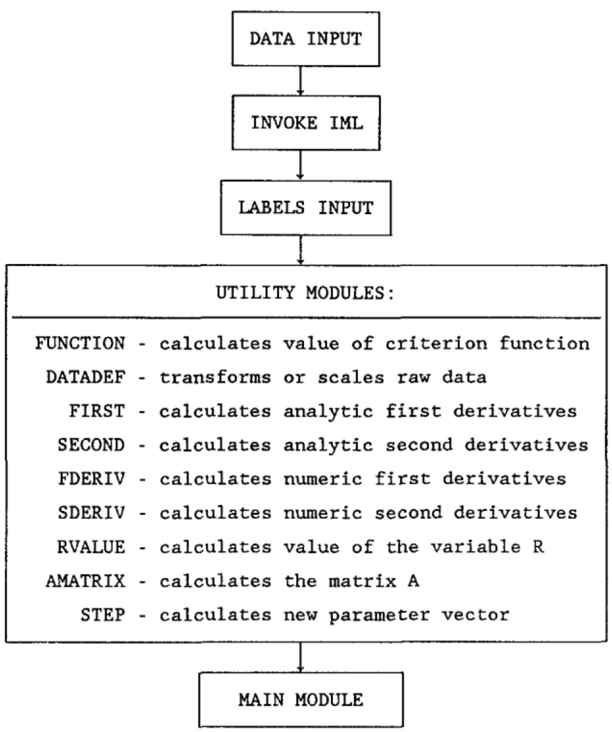

Fig. 1. Structure of EZClimb program.

Table III presents the results of estimating this model with EZClimb along with the results of Chow and of Goldfeld and Quandt. Using numerical derivatives, results are close to Goldfeld and Quandt's who also used the modified quadratic hill-climbing algorithm and virtually identical to Chow's who used a Newton—Raphson method with analytical derivatives. As with the Savin and White example, the only deviation from EZClimb default values was to increase the rate with which *h* was increased.

4. Implementation of EZClimb

Figure 1 provides a schematic overview of EZClimb's code structure. The beginning of the program identifies the location of external data and control files and inputs them into the program. After invoking the IML command (PROC IML), the program then defines a few labels used in the output process. Following this are a number of utility modules which perform much of the detailed calculation required. The module FUNCTION is used to define the criterion function (for example, a likelihood function) to be maximized and is written by the user. (Minimization problems can be solved by defining the criterion function in EZClimb to be the negative of the actual criterion function.) Other modules which may or may not be of concern to the user are:

- DATADEF — transforms or scales data
- FIRST— calculates analytical first derivatives
- SECOND — calculates analytical second derivatives
- DEPVAR — identifies the dependent variable and calculates predicted values of that variable for use in a statistical analysis.

There are five other modules (FDERIV, SDERIV, RVALUE, AMATRIX, and STEP) which need no modification. These calculate numerical derivatives and perform the calculations needed to perform the hill-climbing routine.

As example, consider the code used to estimate Klein's Model I. Because numerical derivatives were used, the insertion of code into the EZClimb program required:

- the location and contents of the external files (in the DATA section)
- the creation of labels (in the LABELS section)
- data conversion to deviation-from-mean form (in the module DATADEF)
- definition of the likelihood function (in the module FUNCTION).

After specifying the location and contents of the external files and creating the desired labels, the module DATADEF was used to transform the data into deviation-from-means form. The code for that module was:

```
START DATADEF(D,OTH);
        UNITCOL-SHAPE(1.0,21,1);
        D-D-UNITCOL*D[:,];
        FINISH;
```

where UNITCOL is a vertical vector of ones and D[:,] creates a horizontal vector whose elements are the means of the variables. Notice that D is both the matrix of *original* observation data (input to the module) as well as the matrix of *transformed* observation data (output from the module). OTH, the vector of constant-across-observations data, is not used for this problem.

Finally, the module FUNCTION was used to define the likelihood function (Equation (19) above). The code for that module was:

```
START FUNCTION(FNVAL,D,OTH,P);* <============ DO NOT ALTER OR REMOVE;
   NEVAL=NEVAL+1;* <======================= DO NOT ALTER OR REMOVE;
   Y=D[,1:3];
   X=D[,4:10];
   BETA12=P[1];
   BETA13=P[2];
   GAMMA12=P[3];
   BETA21=P[4];
   GAMMA24=P[5];
   GAMMA27=P[6];
   BETA31=P[7];
   GAMMA32=P[8];
   GAMMA33=P[9];
   A=SHAPE(0.0,7,3);
   A[1,1]=BETA12;
   A[2,1]=GAMMA12;
   A[3,1]=-BETA13;
   A[5,1]=-BETA13;
   A[6,1]=BETA13;
   A[4,2]=GAMMA24;
   A[5,2]=BETA21;
   A[7,2]=GAMMA27;
   A[2,3]=GAMMA32;
   A[3,3]=GAMMA33;
   B=SHAPE(0.0,3,3);
   B[1,1]=-1.0;
   B[2,1]=BETA12;
   B[3,1]=BETA13;
   B[1,2]=BETA21;
   B[2,2]=-1.0;
   B[1,3]=BETA31;
   B[3,3]=-1.0;
   U=Y*B+X*A;
   DETU2=DET(U'*U);
   IF DETU2 <= 0 THEN DETU2=1.0;
DETB=DET(B);
FNVAL=-0.5*LOG(DETU2/21)+LOG(-DETB);
FINISH;* <=========================== DO NOT ALTER OR REMOVE;
```

Notice that P represents the vector of parameters of the model and that most of this module maps the P vector into a more readable code and defines the matrices A and B. Given those matrices, the definition of the likelihood function is located in the five lines before the FINISH command. Note also that the value of the likelihood function must be set equal to the variable FNVAL.

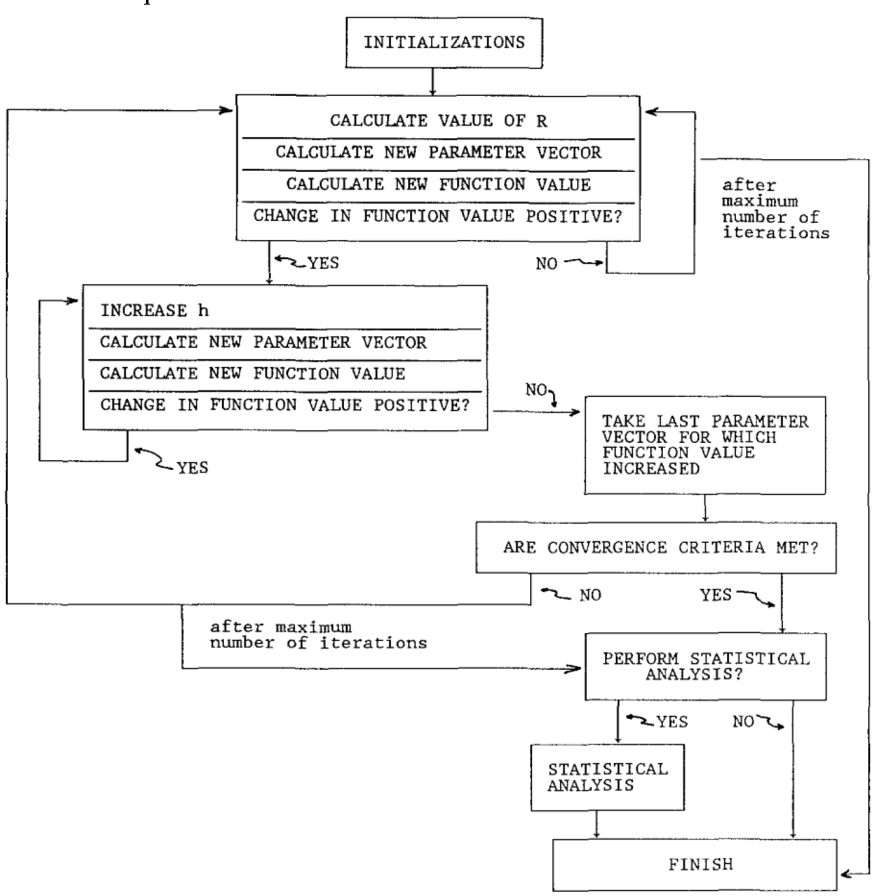

Fig. 2. Logical structure of the MAIN module.

Once the desired code is inserted into the modules of interest, the MAIN module provides the overall guidance for implementing the modified quadratic hill-climbing method. Figure 2 provides a schematic representation of the logical structure of this module. After some initializations and conversion of the data into matrix form (with optional manipulations through the module DATADEF), the MAIN module begins by calculating a value for *R* using the module RVALUE and the old parameter vector **pt.** A tentative set of new parameter values associated with that *R* are then calculated using the modules AMATRIX and STEP. If the tentative set of new parameter values increases the value of the criterion function calculated with the module FUNCTION, the program moves on to determining whether a larger step in the same direction would improve the function still more. If the tentative set of new parameter values does not result in a greater criterion function value, *R* is recalculated and the process is repeated.

Step length is easily calculated. If the tentative set of new parameter values results in a greater criterion function value, the step is multiplied by the positive scalar h(>1) and a second tentative set of new parameter values is generated. If the criterion function increases with this second tentative set of new parameter values, *h* is made still larger. When the criterion function no longer increases, the last step for which an increase occurred is used to define the new parameter vector $\mathbf{p}_{t+1}$. The MAIN module then tests whether the convergence criteria (chosen before the program was run) have been met. See Table IV If the convergence criteria have not been met, the new parameter vector becomes the old parameter vector, and the program begins again to the search for a vector of parameter values which increases the criterion function. If the convergence criteria are met, the program performs an optional statistical analysis using the module DEPVAR before stopping. Throughout this process, information about each iteration is sent to an output file. At the end, summary information on the final parameter values and convergence status is printed.

Table IV. Convergence criteria[a]

Name	Description
FNTOL	proportional change in criterion function less than tolerance value.
PTOL	largest (absolute value) proportional change in criterion function parameter values less than tolerance value.
GTOL	largest (absolute value) element of gradient less than tolerance value.
FETOL[bc]	largest (absolute value) element of criterion-function elasticities with respect to criterion-function parameters less than tolerance value.
SGTOL[bd]	absolute value of step gradient less than tolerance value.

[a] Convergence criteria chosen must be satisfied two times in a row before program will declare convergence and stop. See Belsley (1980, pp. 209–212) and Cramer (1986, p. 73) for a general discussion of convergence criteria.
[b] Not affected by scaling of data.
[c] See Cramer (1986, p. 75) for a fuller description.
[d] Criterion based on conversation with William Greene, Graduate School of Business Administration, New York University. See also Belsley (1980, p. 211).

Two preliminary steps are needed before the program can be submitted for execution. First, the user must insert the desired code into the program to locate external files and define the criterion function. Secondly, the user must create a few external files which provide data and control the program:

- START— contains parameter starting values, that is, $\mathbf{p_o}$ (REQUIRED)
- CONTROL — controls the operation of the program (REQUIRED)
- RAWDAT1— contains data by observation (OPTIONAL)
- RAWDAT2— contains data constant across observations (OPTIONAL).

Note that while the two raw data files are optional, RAWDAT1 will in all likelihood be used. The purpose of RAWDAT2 is to allow the user to input ancillary data such as, for example, a tax rate used in computing disposable income or weights to be used in scaling the data in RAWDAT1. Note also that the names of the files above are those used internally by EZClimb. Externally, the user can give them whatever names are desired.

Table V. Control file description[a]

Variable	Description	Default value
Line 1: Basic variables		
(1) DOBS	Data by observation?	(YES = 1, NO = 0)
(2) DSUP	Data constant across observation?	(YES = 1, NO = 0)
(3) WTS	Transforming or scaling data?	(YES = 1, NO = 0)
(4) STAT	Statistical analysis?	(YES = 1, NO = 0)
(5) ANAL	Analytic derivatives?	(YES = 1, NO = 0)
Line 2: Convergence criteria		
(1) ITER	Iteration limit	100
(2) FNTOL	Change in criterion-function tolerance	1.0E–04
(3) PTOL	Change in parameters tolerance	1.0E–04
(4) GTOL	Gradient tolerance	1.0E–04
(5) FETOL	Function-elasticity tolerance	1.0E–04
(6) SGTOL	Step-gradeint tolerance	1.0E–06
(7) CRIT	Convergence criteria used	7
1 = FNTOL 2 = PTOL 3 = GTOL	4 = FNTOL & PTOL 5 = FNTOL & GTOL 6 = PTOL & GTOL 7 = FNTOL, PTOL, & GTOL 8 = ANY ONE OF ABOVE 9 = ANY TWO OF ABOVE	10 = FETOL 11 = SGTOL
Line 3: Hill-climbing variables		
(1) R	R starting value	1
(2) RCl	R adjustment constant 1	4.0
(3) RC2	R adjustment constant 2	0.4
(4) RITER	R iteration limit	20
(5) H	h starting value	1.0
(6) HFACTOR	h increasing factor	1.1
(7) BETA	β starting value	0.9
(8) EPSILON	value of β's ϵ	0.5
Line 4: Numeric-derivative variables		
(1) DELTA	Proportional change in parameter	1.0E–06
(2) DMIN	Minimum absolute change in parameter	1.0E–08

[a] To choose default value, set variable equal to zero.

Table V provides a summary of the contents of the control file. Line 1 provides information on which files are to be used, whether the program is to modify the raw data input (that is, whether to run the module DATADEF), whether a statistical analysis is desired, and whether analytical derivatives are to be used (that is, whether to run the modules FIRST and SECOND). Line 2 provides information on which set of convergence criteria are to be used and at what tolerance levels. See Table IV for a summary of the various convergence criteria. Line 3 establishes particular values for the several variables used in the hill-climbing routine. Line 4 provides similar information for the variables used in the calculation of numerical derivatives. Default values are provided for lines 2, 3, and 4 and can be invoked by using 0 for the variable's value. All variables in the control file must be specified whether used or not. Having written the necessary code and created the desired external files, the program is submitted to SAS in the usual way.

5. Conclusion

The description of EZClimb and illustration of its efficacy is intended to make more sophisticated estimation methods available to a broader range of re-searchers. To be sure, there are now a large number of software packages available. Their ability to deal with non-standard model forms, however, is limited for those not conversant with Fortran (or other less accessible languages) or not willing to tinker with the internal structure of software. EZClimb may provide a safer alternative.* Its relatively transparent structure and its use of matrix notation make it both flexible and accessible.

Appendix: The Modified Quadratic Hill-Climbing Algorithm

Following Goldfeld, Quandt, and Trotter (1966, 1968), the modified quadratic hill-climbing algorithm is defined to be:

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \begin{cases} -h(\mathbf{S}_t - \alpha\mathbf{A}_t)^{-1}\mathbf{F}_t & \text{if } \mathbf{F}_t \neq 0 \quad \text{and } \alpha > 0 \\ -\mathbf{S}_t^{-1}\mathbf{F}_t & \text{if } \mathbf{F}_t \neq 0 \quad \text{and } \alpha \leqslant 0 \\ +\lambda_1\mathbf{U}_1 & \text{if } \mathbf{F}_t = 0 \quad \text{and } \mathbf{S}_t \text{ not n.d.} \end{cases}$$

where $\mathbf{p}_t$ = vector of parameters

h = step-size adjustment factor (initial value 1)

$\mathbf{S}_t$ = matrix of second derivatives of criterion function

$\mathbf{F}_t$ = vector of first derivatives of criterion function

λ_1 = largest eigenvalue of $\mathbf{S}_t$

$\mathbf{U}_1$ = eigenvector associated with λ_1

$\alpha = \lambda_1 + R(\mathbf{F}_t^{\mathsf{T}}\mathbf{F}_t)^{0.5}$

$$R_{n+1} = R_n \cdot \begin{cases} c_1 & \text{if } Z \leqslant 0 \text{ or } 2 \leqslant Z \\ c_1 + ((c_2 - c_1)/0.7)Z & \text{if } 0 \leqslant Z \leqslant 0.7 \\ c_2 & \text{if } 0.7 \leqslant Z \leqslant 1.3 \\ c_1 - 2((c_1 - c_2)/0.7) + ((c_1 - c_2)/0.7)Z & \text{if } 1.3 \leqslant Z < 2 \end{cases}$$

Z = ratio of the change in the value of the criterion function and the quadratic approximation of that change

$\mathbf{A}_t = \mathbf{B}_t^{\mathsf{T}}\mathbf{B}_t$

$\mathbf{B}_t = \mathbf{I} + ((\beta - 1)/\delta^T\delta)\delta\delta^T$

$\delta = \mathbf{p}_t - \mathbf{p}_{t-1}$

$$\beta_{n+1} = \begin{cases} 0.9 & \text{if } Z \leqslant 0 \text{ or } 2 \leqslant Z \\ \beta_n + (0.9 - \beta_n)C & \text{if } 0 < Z < 2 \text{ and } 0 \leqslant C \\ \beta_n - (0.1 - \beta_n)C & \text{if } 0 < Z < 2 \text{ and } C < 0 \end{cases}$$

$C = (Z - 1)^2 - \epsilon$

c_1, c_2, and ϵ are constants set by user (see Table V for default values).

• Of course, rational choice of software requires that these advantages be balanced against potential monetary costs to the user. SAS/IML is not included in the basic SAS package but instead requires additional fees.

Notes:

[1] The working paper `EZClimb: SAS/IML coding to accompany "Modified quadratic hill-climbing with SAS/IML"' contains a copy of the SAS steps and is available from the author upon request. The SAS steps are also available on floppy disk (for a small fee to cover postage and floppy costs) or via BITNET in ASCII text. The author's BITNET address is leyden@uncg.

[2] Goldfeld, Quandt, and Trotter (1968) refer to the method as *improved* quadratic hill-climbing, while Goldfeld and Quandt (1972) refer to it as *modified* quadratic hill-climbing. I have chosen the latter appellation because of its later date.

[3] If α is defined so that (S_t — αI*)* is negative definite, then $\mathbf{p}_{t+1}$ will be the parameter vector which maximizes the quadratic approximation of the criterion function over a sphere centered at $\mathbf{p}_t$**.** In order to ensure that (S_t — αI*)* is negative definite, α must be no less than λ_1, the largest eigenvalue of S,. See Lemmas 1 and 2, the Theorem, and the Appendix to Goldfeld, Quandt, and Trotter (1966).

[4] The quote can be found in both Goldfeld, Quandt, and Trotter (1968, p. 5) and Goldfeld and Quandt (1972, page 8). Goldfeld, Quandt, and Trotter provide a detailed discussion of the derivation of A_t. Goldfeld and Quandt provide a more concise statement of that derivation.

[5] It should be noted that Berndt, Hall, Hall, and Hausman (1974) argued for the BHHH algorithm on the basis of guaranteed convergence and that they suggest the use of likelihood-ratio testing rather than the calculation of /maximum-likelihood standard errors.

[6] In the three problems examined in this paper, the numerical derivative modules performed well. It should be noted, however, that such success is a function of both the problem being examined and the parameters used to

calculate the numerical derivatives. Particularly when first derivatives are close to zero and/or the second-derivative matrix is close to being singular, the user may be better off using analytic derivatives.

References

Amemiya, T., 1983, Non-linear regression methods, in Z. Griliches and M. D. Intriligator (eds), *Handbook of Econometrics,* Volume 1, North-Holland, New York, pp. 333-389.

Belsley, D. A., 1974, Estimation of systems of simultaneous equations and computational specifications of GREMLIN, *Annals of Economic and Social Measurement 3,* 551-614.

Belsley, D. A., 1979, On the computational competitiveness of full-information maximum-likelihood and three-stage least-squares in the estimation of nonlinear, simultaneous-equations models, *J. Econometrics* **9,** 315-342.

Belsley, D. A., 1980, On the efficient computation of the nonlinear full-information maximum-likelihood estimator, *J. Econometrics* **14,** 203-225.

Berndt, E. K., Hall, B. H., Hall, R. E., and Hausman, J. A., 1974, Estimation and inference in nonlinear structural models, *Annals of Economic and Social Measurement*, **3/4,** 653-665. Chow, G. C., 1983, Nonlinear models, chapter 7 in *Econometrics,* McGraw-Hill, New York. Chow, G. C., 1968, Two methods of computing full-information maximum likelihood estimates in simultaneous stochastic equations, *Internat. Economic Reivew* **9,** 100-112.

Cramer, J. S., 1986, *Econometric Applications of Maximum Likelihood Methods,* Cambridge University Press, New York.

Goldfeld, S. M., and Quandt, R. E., 1972, *Nonlinear Methods in Econometrics,* North-Holland, New York.

Goldfeld, S. M., Quandt, R. E., and Trotter, H. F., 1966, Maximization by quadratic hill-climbing, *Econometrica* **34,** 541-551.

Goldfeld, S. M., Quandt, R. E., and Trotter, H. F., 1968, Maximization by modified quadratic hill-climbing and other methods, Research Memorandum No. 95, Econometric Research Program, Princeton University.

Greene, W., 1990, An introduction to nonlinear optimization, chapter 12 in *Econometric Analysis,* Macmillan, New York.

Judge, G. G., Hall, R. C. Griffiths, W. E., Lutkepohl, H., and Lee, T., 1982, Nonlinear regression models, chapter 24 in *Introduction to the Theory and Practice of Econometrics,* Wiley, New York.

Judge, G. G., Hall, R. C., Griffiths, W. E., Lutkepohl, H., and Lee, T., 1985, Numerical optimization methods, appendix B in *The Theory and Practice of Econometrics,* 2nd edn, Wiley, New York.

Kmenta, J., 1986, Nonlinear models, in *Elements of Econometrics,* 2nd edn, Macmillan, New York, section 11-3.

Leyden, D. **P.,** 1990, EZClimb: SAS/IML coding to accompany 'Modified quadratic hill-climbing with SAS/IML', Working Paper, Department of Economics, University of North Carolina at Greensboro, Greensboro, North Carolina.

Maddala, G. S., 1977, *Econometrics,* McGraw-Hill, New York, pp. 171-181.

Moffitt, R., 1986, The econometrics of piecewise-linear constraints: a survey and exposition of the maximum likelihood method, *J. Business and Economic Statistics* **4,** 317-328.

Powell, M. J. D., 1970, A survey of numerical methods for unconstrained optimization, *Soc. Industrial and Applied Mathematics Rev.* **12,** 79-97.

Quandt, R. E., 1983, Computational problems and methods, chapter 12 in Z. Griliches and M. D. Intriligator, (eds), *Handbook of Econometrics,* Volume 1, North-Holland, New York.

Rosenbrock, H. H., 1961, An automatic method for finding the greatest or least value of a function, *Computer J., 3,* 175-184.

SAS Institute Inc., 1985, *SASIIML User's Guide, Version 5 Edition,* SAS Institute, Inc., Cary, North Carolina.

Savin, N. E., and White, K. J., 1978, Estimation and testing for functional form and autocorrelation, *J. Econometrics* 8, 1-12.

Spitzer, *J. J.,* 1984, Variance estimates in models with the Box-Cox transformation: Implications for estimation and hypothesis testing, *Rev. Economics and Statistics* 66, 645-652.