This document, first drafted and circulated in January 2007, was supported by the Natural Sciences and Engineering Council of Canada (NSERC) and the *Fonds Québécois de Recherche sur la Nature et les Technologies* (FQRNT). It complements the estimation procedures described in: *Tran, C.L. and M. Gaudry, "The L-2.1 estimation procedures for BC-DAUHESEQ (Box-Cox Directed AUtoregressive HEteroskedastic Single EQuation) regression". Publication AJD-108, Agora Jules Dupuit, Université de Montréal, 26 p., April 2008.*

PUBLICATION
AJD-109

LEVEL: The L-2.1 program user guide with databases, examples and Tablex outputs

Cong-Liem Tran[1] and Marc Gaudry[1,2]

[1]Agora Jules Dupuit, Université de Montréal.
[2]Département de sciences économiques, Université de Montréal.
(*B.P. 6128 Succursale centre ville, Montréal, QC, Canada H3C 3J7*)

Publication AJD-109, Agora Jules Dupuit
Université de Montréal
April 2008, revised October 2009, June 2011, November 2012, May 2014

Table of Contents

1. LEVEL L-2.1 PROGRAM

The LEVEL L-2.1 program is designed to deal with the specification of the functional form in the generalized single-equation regression where the funtional form of heteroskedasticity of the residuals, which may also be autocorrelated using an *R*-Koyck process for each autoregressive order considered, can be estimated. This process is the analog of the usual Koyck transformation on a time series, but extended to a matrix *R* which is called a contiguity matrix in the context of spatial econometrics and which can be constructed according to any set of rules defined or *directed* by the analyst. For each order of correlation ℓ, the *R*-Koyck process only adds to the usual autoregressive parameter ρ_ℓ a proximity parameter π_ℓ which measures the relative weight of close and remote neighbors in an infinite distributed lag structure. A maximum likelihood procedure is used to estimate jointly all the parameters including the regression coefficients, the coefficients in the heteroskedasticity function, the autocorrelation and proximity parameters as well as those of the Box-Cox transformations on the dependent, independent and heteroskedasticity variables.

The asymptotic covariance matrix of all the parameter estimates in a regression equation is computed. A number of useful outputs includes elasticities of the dependent variable with respect to the independent variables and marginal rates of substitution between two independent variables.

The algorithm used in the maximum likelihood estimation and the different outputs of the L-2.1 program are described in *Tran, C.L. and M. Gaudry, LEVEL: The L-2.1 program for BC-DAUHESEQ (Box-Cox Directed AUtoregressive HEteroskedastic Single Equation) regressions, Publication AJD-108, Agora Jules Dupuit,* Université de Montréal, 2008. It can be downloaded from the Agora Jules Dupuit web site currently at www.e-ajd.net.

1.1 Installing Level L-2.1 and Tablex programs

The file *Level2new.zip*, which can be downloaded from the website of Agora Jules Dupuit currently found at www.e-ajd.net, contains all the folders used in this publication. To install the two programs with their examples of input/output files and the associated databases, unzip *Level2new.zip* to drive *d:*.

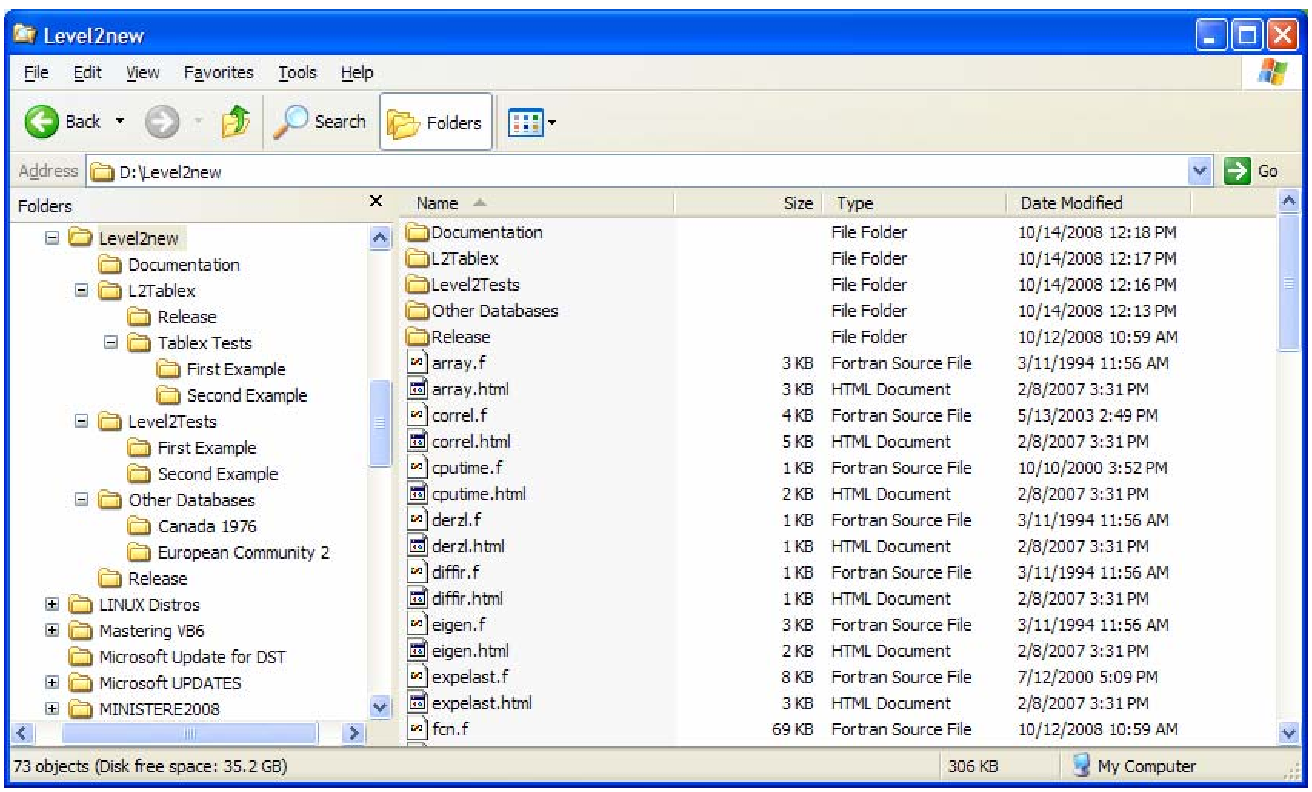
The *d:\Level2new* folder will be created with the following structure: in the root folder *d:\Level2new*, all the Fortran files (extension *.f*) and specific files used in the compilation of L-2.1 estimation program with Compaq Visual Fortran 6.6 are stored. Under this folder, there are five subfolders:

- The *Documentation* subfolder contains all the publications cited in the text but not available at the website of Agora Jules Dupuit.
- The *L2Tablex* subfolder includes the specific files used in the compilation of L-2.1 Tablex program with Compaq Visual Fortran 6.6. The compiled file *L2Tablex.exe* is stored in subfolder *L2Tablex\Release*. The *L2Tablex\Tablex Tests* subfolder contains two Tablex examples with their input/output files. These tablexes correspond to the variants estimated in the two examples found in the *Level2Tests* subfolder below.
- The *Level2Tests* subfolder includes two examples of L-2.1 estimation with their input/output files and associated databases:

1. The first example reproduces the estimation of six variants reported in AJD-27 (*Gaudry* (2004)), Table NORTH AMERICA 2, p. 12. The data file *America.in* and the contiguity matrix *R400miles.in* (DIST_OD) are stored in *D:\Level2new\Level2Tests\First Example*.
2. The second example reproduces the estimation of six variants reported in AJD-14 (*Gaudry et al.* (1994, 2008)), Table B.2 (CANADA 1976), p. 51. The data file *Canada1976.in* and the contiguity matrix *R320km.in* (DIST_OD) are stored in *D:\Level2new\Level2Tests\Second Example*.

- The *Other Databases* subfolder contains two additional subfolders not used in the two examples above, but used for two other tables reported in AJD-14 and AJD-27:
  1. *Canada 1976* where the data file *Canada1976.in* and the associated contiguity matrices *R1.in* (DIST_O) and *R2.in* (DIST_D) for the first and second orders of autocorrelation are used in the estimation of six variants reported in AJD-14, Table B.3 (CANADA 1976), p. 52. Note that the data file *Canada1976.in* is also used for the second example above.
  2. *European Community 2* where the data file *Europe.in* and the associated contiguity matrix *R300km.in* (DIST_OD) are used in the estimation of six variants reported in AJD-27, Table EUROPEAN COMMUNITY 2, p. 13.
- The *Release* subfolder contains the compiled file *level2.exe* of L-2.1 estimation program.

The L-2.1 estimation program can run only on a Pentium 4 with at least 1 G. of ram due to the inversion of big matrices involving the contiguity matrices R_1 and R_2 whose dimensions are n by n where n is the maximum number of observations fixed at 1000.



1.2 Input File

The input file for LEVEL 2.1 program contains five sections where the Fortran Namelist $…$ is used to input the values of the parameters.

SECTION	INPUT CONTENTS
I	
Control Parameters	**_$model_nlist= ,nstep= ,epsi= ,wght= ,nobs= ,namy=,nvaxz= ,nvax= ,norder= , nvaz= ,nlamz= , _ nlamy= ,nlamx= ,nscale= ,nexpy= ,nres= ,ncor= ,ncov= , varname= ,nvariant= $** (Symbol _ indicates a blank and an input value for each parameter must be specified after the equality sign.)
II	
Directed Autocorrelation	**_$autocor_rho= , pi= ,ntpi= ,matfile= $** (This card is necessary if **norder** = 1 or 2.)
III	
Heteroskedasticity	**_$hetero_numz= ,numlz= ,delta= ,ntdel= ,vlz= ,ntlz= $** (This card is necessary if **nvaz** > 0.)
IV	
Box-Cox on the Dependent and Independent Variables	**_$yboxcox_vly= ,ntly= ,numlxy= $** (This card is necessary if **nlamy** = 1.) **_$xboxcox_numx= ,numlx= ,vlx= ,ntlx= $** (This card is necessary if **nlamx** > 0.)
V	
Data	**_$yxzdata_yxzfile= ,yxzfmt= $**

1.3 Section I : Control Parameters

nlist	1 – List the iterations during the maximization of the concentrated log-likelihood function. 0 – No list.
nstep	Maximum number of steps for the maximization of the concentrated log-likelihood function. Usually, nstep can be set at least equal to the square of the number of the estimated parameters in the concentrated log-likelihood function.
epsi	Convergence parameter for the concentrated log-likelihood function ($10^{-7} \leq$ epsi $\leq 10^{-4}$).
wght	Common weight for the relative changes in the values of the estimated parameters during the maximization of the concentrated log-likelihood function. Usually, wght=1, but smaller values such as 0.1, 0.01 or 0.001 can be tried if there is a problem of convergence.
nobs	Total number of observations for estimation (max. 1000).
namy	Name of the dependent variable Y within apostrophes,used for estimation.
nvaxz	Total number of dependent, independent variables and heteroskedasticity variables in the data file declared in Section V.
nvax	Number of independent variables X used in estimation, excluding the constant which is generated in the program.
norder	Number of directed autocorrelation orders used in estimation ($0 \leq$ norder ≤ 2).
nvaz	Number of heteroskedasticity variables Z used in estimation ($0 \leq$ nvaz ≤ 10).
nlamz	Number of distinct λ_z 's (fixed and estimated) used in estimation ($0 \leq$ nlamz ≤ 10).
nlamy	1 – One Box-Cox transformation on the dependent variable Y. 0 – No Box-Cox transformation on Y.

nlamx	Number of distinct λ_x 's (fixed or estimated) used in estimation ($0 \leq$ nlamx ≤ 10).
nscale	1 – Scaling of Y , X 's and Z 's. 0 – No scaling.
nexpy	Number of observed, expected values of Y and their differences, and sample Y to be listed for $0 \leq$ nexpy $\leq$ nobs at the initial and final iterations.
nres	Number of estimated residuals u , v and w to be listed ($0 \leq$ nres $\leq$ nobs).
ncor	2 – Table of Variance-Decomposition Proportions for the independent variables X (including the Constant) and Correlation Matrix for the independent variables X (excluding the Constant) and the dependent variable Y . Both tables are computed and printed for the original and transformed variables. 1 – Both tables are computed and printed for the transformed variables only. 0 – No print for both tables. Only the correlation matrix for X is computed for TABLEX program which needs the results.
ncov	1 – Print the covariance matrix of the estimated parameters at the maximum point of the log-likelihood function. 0 – No print.
varname	Name of the estimated variant (max. 9 characters). Choose a name to help you remember roughly about the specification of the variant for the Box-Cox (BC) transformations, autocorrelation (AU) or heteroskedasticity (HG). For example, BC3_AUHG indicates that there are three Box-Cox transformations in total among the lambda(y) and lambda(x) either fixed or estimated, and that autocorrelation and heteroskedasticity are considered in the variant estimation.
nvariant	Number of the estimated variant (max. 3 digits). The variant number should start at 1 and should be unique.
Notes	For a normal run when there is at least one parameter to be estimated in the concentrated log-likelihood function other than the β -coefficients, namely λ_y , λ_x , δ , λ_z , ρ or π , The estimated results are stored in a file for later use in TABLEX program. For a special run when there is no other parameter to be estimated than the β -coeffi-cients, i.e. when there is no autocorrelation and all the λ_y , λ_x , δ and λ_z are either fixed or not considered, the β -coefficients are just estimated by an ordinary least squares regression, given the fixed values of all the other parameters if any. The estimated results are stored in a file for later use in TABLEX program.

1.4 Section II : Directed Autocorrelation

rho	Vector of initial values for ρ_1 and ρ_2 . If **norder** = 1, only one value for ρ_1 is specified. If **norder** = 2, two values for ρ_1 and ρ_2 , separated by a comma, are specified. Note that the ρ 's cannot be fixed but estimated only.
pi	Vector of fixed or initial values for π_1 and π_2 . If **norder** = 1, only one value for π_1 need to be specified. If **norder** = 2, two values for π_1 and π_2 , separated by a comma, are specified. Note that the π 's can be fixed or estimated.
ntpi	Vector of types of π_1 and π_2 . A type value of 0 or 1 indicates that a π is fixed or estimated respectively. If **norder** = 1, only one type value for π_1 is specified. If **norder** = 2, two type values for π_1 and π_2 , separated by a comma, are specified.
matfile	Vector of two file names (at most) for the contiguity matrices R_1 and R_2 of the first and second autocorrelation orders respectively. Each name should appear within apostrophes and the two names are separated by a comma: for example, matfile='name1','name2' if there are two autocorrelation orders. If only one order is considered, then matfile='name1'.

1.5 Section III : Heteroskedasticity

numz	Vector of index numbers of the heteroskedasticity variables Z used in estimation.
numlz	Vector of index numbers of the fixed or estimated λ_z 's associated with the heteroskedasticity variables Z used in estimation. If a variable is not transformed by Box-Cox, then its associated **numlz** is set at 0.
delta	Vector of fixed or initial values of the δ 's.
ntdel	Vector of types of the δ 's. If a δ is fixed (or estimated), then its associated **ntdel** is set at 0 (or 1).
vlz	Vector of fixed or initial values of the λ_z 's.
ntlz	Vector of types of the λ_z 's. If a λ_z is fixed (or estimated), then its associated **ntlz** is set at 0 (or 1).

1.6 Section IV: Box-Cox on the Dependent and Independent Variables

vly	Fixed value (if **ntly**=0) or initial value (if **ntly**=1) of λ_y associated with the dependent variable Y .
ntly	Type of λ_y . If λ_y is fixed (or estimated), then **ntly** = 0 (or 1).
numlxy	Index number of a λ_x associated with one or several independent variables X such that λ_x is constrained to be equal to λ_y when both Box-Cox parameters are estimated.
numx	Vector of index numbers of the independent variables X used in estimation.
numlx	Vector of index numbers of the fixed or estimated λ_x 's associated with the independent variables X used in estimation. If a variable X is not transformed by Box-Cox, then its associated **numlx** is set at 0.
vlx	Vector of fixed or initial values of the λ_x 's.
ntlx	Vector of types of the λ_x 's. If a λ_x is fixed (or estimated), then its associated **ntlx** is set at 0 (or 1).

1.7 Section V: Data

yxzfile	Name of the data file within apostrophes, containing the vector of all the variable names (enter one name per line) followed by the observations on the dependent, independent and heteroskedasticity variables. The data are entered by observation, i.e. each observation contains Y , X and Z in any order.
yxzfmt	Fortran input format within parentheses, then within apostrophes, for the observations on Y , X and Z .

1.8 Running L-2.1 Program

Since L-2.1 program is a DOS-program, it can only be run in a DOS dialog box:

1. Select START/RUN menu and type the command **cmd** to open the DOS dialog box.
2. At the DOS prompt, change to the directory where the input file and the database are stored.
3. To run the program, type the command:
   d:\level2new\release\level2.exe < input file name > output file name.

The output file associated with an estimated variant will be stored in the same directory.
An easier way to run the program without the DOS dialog box is to store the command in a batch file (extension .bat) and double-click on this file to execute the command. We will use this method to

estimate the variants for the two examples below. For the variant estimation procedures, see *Tran and Gaudry (2008)*.

1.9 First Example of Input Files

The first example can be found in directory *d:\Level2new\Level2Tests\First Example*. It illustrates the estimation of the six variants reported in Table **NORTH AMERICA 2: TRADE FLOWS BETWEEN 30 AMERICAN STATES & 10 CANADIAN PROVINCES, AND AMONG THE LATTER (1988 SHIPMENTS AT PRODUCER PRICES)** of *Gaudr*y (2004), p. 12. The corresponding batch files (.bat) are also stored in this directory.

The data file *America.in* contains 683 O-D pairs (observations). Since the whole file is too big to be reproduced here, only the first three observations are listed below. The first variable is the dependent variable SHIP (1988 shipments at producer prices), followed by eight independent variables whose definitions will be given later in section 2.3: First Example of Input File for L-2.1 Tablex. Only four independent variables (AGDPM, AGDPX, CDUMMY, DIST) are used in the regressions and the last variable DPMDPX is used in the heteroskedasticity function $f(Z_t)$ for variant 6. The Fortran format for the observations is (5e15.8): it indicates that 5 values are entered on a single line and that the field for each value is 15 characters. Each value entered is right-justified and contains 8 decimals after the point followed by the exponential of the form E+XX or E-XX. For those unfamiliar with the Fortran format specifications, a good Fortran textbook should be consulted.

Data file: America.in

```
SHIP
AG
AGDPM
AGDPX
CDUMMY
DIST
EXPORT
MFG
DPMDPX
 0.69023373E+05 0.22256900E-01 0.12253200E+03 0.64059000E+02 0.00000000E+00
 0.10850000E+04 0.69026203E+05 0.68752497E-01 0.78492774E+04
 0.68507590E+05 0.13988700E-01 0.12253200E+03 0.62375000E+02 0.00000000E+00
 0.21830000E+04 0.68507703E+05 0.36321301E-01 0.76429332E+04
 0.72948941E+06 0.86272396E-02 0.12253200E+03 0.64230900E+03 0.00000000E+00
 0.24630000E+04 0.72950200E+06 0.77743498E-02 0.78703403E+05
```

Contiguity matrix R_1 file

Except for the first variant, the first-order autocorrelation is considered in the others. The file for the contiguity matrix R_1 is *R400miles.in* . Since the matrix is very big (dimension 683 x 683), only the first row is listed below. Note that the name of the matrix (max. 8 characters) should appear on the first line.

In creating this matrix, a rule is used to specify the contiguity or proximity for each O-D pair: if the distance between the origin and the destination of a pair is less than 400 miles, then a value of 1 indicating that they are neighbors is assigned to the pair. Otherwise a value of 0 is specified. The Fortran format of the matrix is always (20f3.0) and cannot be changed by the user. It indicates that 20 values are entered on a single line and that the field for each value is 3 characters. Each value entered is right-justified and without a decimal after the point.

```
DIST_OD
 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0.
```

Input file: variant1.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with fixed $\lambda_y = 0$ and one common Box-Cox transformation on the three independent variables AGDPM, AGDPX and DIST with fixed $\lambda_1 = 0$. The fourth independent variable CDUMMY is a real dummy, hence cannot be transformed by Box-Cox.
- No heteroskedasticity nor autocorrelation is considered.
- Display observed, expected values of Y and their differences, and sample Y for 120 observations with option nexpy=120.

```
$model nlist=1,nstep=2000,epsi=1.e-7,wght=1.,nobs=683,namy='SHIP',nvaxz=8,nvax=4,norder=0,
nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=0,nexpy=120,nres=0,ncor=0,ncov=0,varname='LOG',nvariant=1$
$yboxcox vly=0.,ntly=0,numlxy=0$
$xboxcox numx=2,3,4,5,numlx=1,1,0,1,vlx=0.,ntlx=0$
$yxzdata yxzfile='America.in',yxzfmt='(5e15.8)'$
```

Input file: variant2.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with fixed $\lambda_y = 0$ and one common Box-Cox transformation on the three independent variables AGDPM, AGDPX and DIST with fixed $\lambda_1 = 0$. The fourth independent variable CDUMMY is a real dummy, hence cannot be transformed by Box-Cox.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and fixed $\pi_1 = 1$.

```
$model nlist=1,nstep=2000,epsi=1.e-7,wght=1.,nobs=683,namy='SHIP',nvaxz=8,nvax=4,norder=1,
 nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=0,nexpy=0,nres=0,ncor=0,ncov=0,varname='LOG+AU',nvariant=2$
 $autocor rho=0.,pi=1.,ntpi=0,matfile='R400miles.in'$
 $yboxcox vly=0.,ntly=0,numlxy=0$
 $xboxcox numx=2,3,4,5,numlx=1,1,0,1,vlx=0.,ntlx=0$
 $yxzdata yxzfile='America.in',yxzfmt='(5e15.8)'$
```

Input file: variant3.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with fixed $\lambda_y = 0$ and one common Box-Cox transformation on the three independent variables AGDPM, AGDPX and DIST with fixed $\lambda_1 = 0$. The fourth independent variable CDUMMY is a real dummy, hence cannot be transformed by Box-Cox.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.e-7,wght=1.,nobs=683,namy='SHIP',nvaxz=8,nvax=4,norder=1,
 nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='LOG+AU+PR',nvariant=3$
 $autocor rho=0.,pi=0.99,ntpi=1,matfile='R400miles.in'$
 $yboxcox vly=0.,ntly=0,numlxy=0$
 $xboxcox numx=2,3,4,5,numlx=1,1,0,1,vlx=0.,ntlx=0$
 $yxzdata yxzfile='America.in',yxzfmt='(5e15.8)'$
```

Input file: variant4.in

The specification for this variant is as follows:

- One common Box-Cox transformation on the dependent variable Y and the three independent variables AGDPM, AGDPX and DIST with estimated λ_1. The fourth independent variable CDUMMY is a real dummy, hence cannot be transformed by Box-Cox.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.e-7,wght=1.,nobs=683,namy='SHIP',nvaxz=8,nvax=4,norder=1,
 nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='BC1+AU+PR',nvariant=4$
 $autocor rho=0.6619374,pi=0.383496,ntpi=1,matfile='R400miles.in'$
 $yboxcox vly=0.,ntly=1,numlxy=1$
 $xboxcox numx=2,3,4,5,numlx=1,1,0,1,vlx=0.,ntlx=1$
 $yxzdata yxzfile='America.in',yxzfmt='(5e15.8)'$
```

Input file: variant5.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with estimated λ_y and one common Box-Cox transformation on the three independent variables AGDPM, AGDPX and DIST with estimated λ_1. The fourth independent variable CDUMMY is a real dummy, hence cannot be transformed by Box-Cox.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.e-7,wght=1.,nobs=683,namy='SHIP',nvaxz=8,nvax=4,norder=1,
 nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='BC2+AU+PR',nvariant=5$
 $autocor rho=0.6619374,pi=0.383496,ntpi=1,matfile='R400miles.in'$
 $yboxcox vly=0.,ntly=1,numlxy=0$
 $xboxcox numx=2,3,4,5,numlx=1,1,0,1,vlx=0.,ntlx=1$
 $yxzdata yxzfile='America.in',yxzfmt='(5e15.8)'$
```

Input file: variant6.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with estimated λ_y and one common Box-Cox transformation on the three independent variables AGDPM, AGDPX and DIST with estimated λ_1. The fourth independent variable CDUMMY is a real dummy, hence cannot be transformed by Box-Cox.
- Heteroskedasticity with the variable DPMDPX whose λ_z is fixed at 0.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.e-7,wght=1.,nobs=683,namy='SHIP',nvaxz=8,nvax=4,norder=1,
 nvaz=1,nlamz=1,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='BC2GHAUPR',nvariant=6$
 $autocor rho=0.650,pi=0.421,ntpi=1,matfile='R400miles.in'$
 $hetero numz=8,numlz=1,delta=0.,ntdel=1,vlz=0.,ntlz=0$
 $yboxcox vly=0.0658,ntly=1,numlxy=0$
 $xboxcox numx=2,3,4,5,numlx=1,1,0,1,vlx=0.0321,ntlx=1$
 $yxzdata yxzfile='America.in',yxzfmt='(5e15.8)'$
```

1.10 First Example of Output Files

The output files (.txt) which contain the complete results of the variant estimations are stored in the same folder as the input files (.in). For each variant, for example *variant1* whose variant name is LOGIT and variant number is 1, two additional output files *LOG1.tab* and *LOGIT1_ELA.txt* are also created in the same folder. The first file *LOG1.tab* contains the essential results which will be used as input to L-2.1 Tablex program whereas the latter contains the conditional t-value of each coefficient β_k and the sample elasticity of the dependent variable Y with respect to each independent variable X_k, evaluated at the sample means and excluding the regression constant. The second file *LOGIT1_ELA.txt* will be used as input to the TABQDF program which combines total and mode choice results of LEVEL L-2.1 and SHARE S1/S5 or PROBABILITY P2/P6 to obtain modal elasticities and diversion rates in the so-called Quasi Direct Format found in *Liem and Gaudry* (1994,1998).

Output file: variant1.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         120,
 NAMY    = TOTFLOW   ,
 NVAXZ   =           4,
 NVAX    =           4,
 NORDER  =           0,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           0,
 NEXPY   =         120,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = LOG      ,
 NVARIANT       =           1
 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           0,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           1,           2,           3,           4, 46*0,
 NUMLX   = 4*1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 10*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                                   ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07 -0.4112D+07

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05 -0.5518D+05

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05 -0.1190D+05

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01 -0.8368D+00

CONSTANT    -0.2432D-06 -0.1812D-04 -0.8406D-04 -0.1195D+01  0.1000D+01
```

	Observed Y	Sample Y
1	0.3865100D+05	0.1694362D+05
2	0.7429900D+05	0.3623194D+05
3	0.5340000D+04	0.5583031D+04
4	0.1116600D+05	0.9655887D+04
5	0.3585000D+04	0.4964320D+04
6	0.8164000D+04	0.6036894D+04
7	0.4554000D+04	0.5062342D+04
8	0.7521000D+04	0.6106623D+04
9	0.6593000D+04	0.5360027D+04
10	0.8782000D+04	0.7911153D+04
11	0.2187600D+05	0.2407340D+05
12	0.1213300D+06	0.1351666D+06
13	0.8127000D+05	0.1319997D+06
14	0.1540400D+05	0.1674094D+05
15	0.2147400D+05	0.2744050D+05
16	0.1228400D+05	0.1588238D+05
17	0.1393900D+05	0.1416564D+05
18	0.1069300D+05	0.1231631D+05
19	0.1030200D+05	0.1173857D+05
20	0.5406000D+04	0.7374671D+04
21	0.4521000D+04	0.7559916D+04
22	0.2127500D+07	0.2362537D+07
23	0.2715900D+06	0.2042691D+06
24	0.1517900D+06	0.1213918D+06
25	0.1521500D+07	0.9396670D+06
26	0.7158900D+05	0.5221128D+05
27	0.1650900D+06	0.1343316D+06
28	0.2963400D+06	0.2276690D+06
29	0.2093500D+06	0.1586046D+06
30	0.1498900D+06	0.1171726D+06
31	0.4746000D+05	0.3841186D+05
32	0.1076100D+06	0.8875091D+05
33	0.6450100D+05	0.4678881D+05
34	0.9202700D+05	0.8908211D+05
35	0.2497100D+05	0.1916053D+05
36	0.3748600D+05	0.4146766D+05
37	0.8678000D+04	0.1301998D+05
38	0.6718000D+04	0.1110976D+05
39	0.3146300D+05	0.2527879D+05
40	0.2328600D+05	0.2586877D+05
41	0.5486100D+05	0.4006595D+05
42	0.2333700D+06	0.2446231D+06
43	0.8903600D+05	0.8075967D+05
44	0.8340500D+06	0.6878364D+06
45	0.3720100D+05	0.3341297D+05
46	0.8879500D+05	0.8476986D+05
47	0.1601100D+06	0.1516421D+06
48	0.1146300D+06	0.9863604D+05
49	0.8406300D+05	0.6634545D+05
50	0.2168200D+05	0.2030727D+05
51	0.5095900D+05	0.4638687D+05
52	0.5377700D+05	0.3983535D+05
53	0.6716700D+05	0.6305910D+05
54	0.1529300D+05	0.1007107D+05
55	0.3626200D+05	0.2248458D+05
56	0.1222400D+05	0.7706285D+04
57	0.9907000D+04	0.6246388D+04
58	0.2310300D+05	0.1263661D+05
59	0.2278700D+05	0.1568218D+05
60	0.3585400D+05	0.2001092D+05
61	0.3893600D+06	0.4160141D+06
62	0.4266300D+05	0.3942551D+05
63	0.7401600D+05	0.7272775D+05
64	0.4274600D+05	0.3740440D+05
65	0.3194800D+05	0.2758536D+05
66	0.8775000D+04	0.7048378D+04
67	0.1715600D+05	0.1516576D+05
68	0.1195300D+05	0.9560486D+04
69	0.2736000D+07	0.5296232D+07
70	0.1289300D+07	0.1752909D+07

71	0.2139400D+06	0.2040199D+06
72	0.5161200D+06	0.4258026D+06
73	0.1494000D+06	0.1177774D+06
74	0.2677600D+06	0.2160699D+06
75	0.8032200D+05	0.4027864D+05
76	0.1010400D+06	0.8772827D+05
77	0.2942800D+05	0.3630256D+05
78	0.2226600D+05	0.2854859D+05
79	0.7673200D+05	0.5561094D+05
80	0.6434800D+05	0.5448193D+05
81	0.1222000D+06	0.7917129D+05
82	0.6370900D+06	0.1055302D+07
83	0.7216000D+05	0.7027974D+05
84	0.1315500D+06	0.1399052D+06
85	0.2025800D+05	0.1508378D+05
86	0.3354000D+05	0.2956421D+05
87	0.5720000D+04	0.1221787D+05
88	0.3154000D+04	0.8179310D+04
89	0.5015000D+04	0.1170637D+05
90	0.7885000D+06	0.1311781D+07
91	0.9269500D+05	0.9527966D+05
92	0.1692700D+06	0.1698514D+06
93	0.3411500D+05	0.2613678D+05
94	0.5428600D+05	0.4980824D+05
95	0.1421600D+05	0.2066731D+05
96	0.8656000D+04	0.1319045D+05
97	0.1496700D+05	0.1873423D+05
98	0.3195000D+06	0.4231540D+06
99	0.4823100D+05	0.4238128D+05
100	0.9668800D+05	0.8790677D+05
101	0.1993900D+05	0.1460522D+05
102	0.4677000D+04	0.7535075D+04
103	0.2577000D+04	0.6078853D+04
104	0.4218000D+04	0.8508376D+04
105	0.1490900D+06	0.1980574D+06
106	0.2599500D+06	0.3255238D+06
107	0.1497300D+05	0.1164204D+05
108	0.2634000D+05	0.2277208D+05
109	0.9546000D+04	0.1372761D+05
110	0.6170000D+04	0.8052637D+04
111	0.9086000D+04	0.1277964D+05
112	0.2581700D+06	0.4860611D+06
113	0.7723000D+04	0.7208375D+04
114	0.9061000D+04	0.7385582D+04
115	0.1740200D+05	0.1631865D+05
116	0.7988000D+04	0.1861496D+05
117	0.3310000D+04	0.8432934D+04
118	0.6790000D+04	0.1215664D+05
119	0.1126800D+06	0.1233341D+06
120	0.1116000D+05	0.6244799D+04

INITIAL ESTIMATES

	LAM	PARAMETER	STD-ERROR	STUDENT-T	GRADIENT	DERIV.Y DERIV.E(Y)	ELAST.Y ELAST.E(Y)
POP	X1	0.1389205D+01	0.5457770D-01	25.4537107	-0.1024327D-05	0.3872510D+00 0.4970914D+00	1.3892049 1.3892049
REVE	X1	0.8322593D+00	0.6932686D+00	1.2004861	-0.6618732D-06	0.2885581D+02 0.3626500D+02	0.8322593 0.8322593

```
LANGFR        X1  0.6643968D-01  0.5550501D-01      1.1970033 -0.3276004D-06  0.1338513D+03
0.0664397
                                                                               0.1916400D+03
0.0664397

UTILITY       X1  0.6335200D+00  0.1496708D-01     42.3275532  0.4788431D-06  0.1902915D+07
0.6335200
                                                                               0.2843971D+08
0.6335200

CONSTANT      X0 -0.1078358D+02  0.5515421D+01     -1.9551693 -0.7920692D-07 -0.1592291D+07    -
10.7835828
                                                                              -0.2006817D+07    -
10.7835828

VAR(W)            0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00


LOG-LIKELIHOOD = -0.1318235D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.8850
PEARSON-R2(E) (adjusted for D.F.)   =        0.8810

PSEUDO-R2(L)  (unadjusted)          =        0.9441
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9422

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+06

MEAN of residual W                  = -0.8662091D-10
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00


         Observed Y     Expected Y  Obs(Y)-Exp(Y)
   1  0.3865100D+05  0.1809265D+05  0.2055835D+05
   2  0.7429900D+05  0.3868900D+05  0.3561000D+05
   3  0.5340000D+04  0.5961643D+04 -0.6216431D+03
   4  0.1116600D+05  0.1031070D+05  0.8553005D+03
   5  0.3585000D+04  0.5300975D+04 -0.1715975D+04
   6  0.8164000D+04  0.6446285D+04  0.1717715D+04
   7  0.4554000D+04  0.5405644D+04 -0.8516444D+03
   8  0.7521000D+04  0.6520742D+04  0.1000258D+04
   9  0.6593000D+04  0.5723516D+04  0.8694838D+03
  10  0.8782000D+04  0.8447647D+04  0.3343529D+03
  11  0.2187600D+05  0.2570593D+05 -0.3829932D+04
  12  0.1213300D+06  0.1443329D+06 -0.2300294D+05
  13  0.8127000D+05  0.1409513D+06 -0.5968126D+05
  14  0.1540400D+05  0.1787623D+05 -0.2472228D+04
  15  0.2147400D+05  0.2930137D+05 -0.7827370D+04
  16  0.1228400D+05  0.1695944D+05 -0.4675443D+04
  17  0.1393900D+05  0.1512628D+05 -0.1187278D+04
  18  0.1069300D+05  0.1315154D+05 -0.2458538D+04
  19  0.1030200D+05  0.1253462D+05 -0.2232615D+04
  20  0.5406000D+04  0.7874783D+04 -0.2468783D+04
  21  0.4521000D+04  0.8072591D+04 -0.3551591D+04
  22  0.2127500D+07  0.2522752D+07 -0.3952520D+06
  23  0.2715900D+06  0.2181216D+06  0.5346845D+05
  24  0.1517900D+06  0.1296240D+06  0.2216604D+05
  25  0.1521500D+07  0.1003390D+07  0.5181096D+06
  26  0.7158900D+05  0.5575198D+05  0.1583702D+05
  27  0.1650900D+06  0.1434413D+06  0.2164867D+05
  28  0.2963400D+06  0.2431084D+06  0.5323162D+05
```

29	0.2093500D+06	0.1693603D+06	0.3998968D+05
30	0.1498900D+06	0.1251186D+06	0.2477136D+05
31	0.4746000D+05	0.4101675D+05	0.6443245D+04
32	0.1076100D+06	0.9476954D+05	0.1284046D+05
33	0.6450100D+05	0.4996178D+05	0.1453922D+05
34	0.9202700D+05	0.9512321D+05	-0.3096206D+04
35	0.2497100D+05	0.2045990D+05	0.4511097D+04
36	0.3748600D+05	0.4427978D+05	-0.6793782D+04
37	0.8678000D+04	0.1390293D+05	-0.5224933D+04
38	0.6718000D+04	0.1186317D+05	-0.5145169D+04
39	0.3146300D+05	0.2699307D+05	0.4469932D+04
40	0.2328600D+05	0.2762305D+05	-0.4337053D+04
41	0.5486100D+05	0.4278302D+05	0.1207798D+05
42	0.2333700D+06	0.2612121D+06	-0.2784214D+05
43	0.8903600D+05	0.8623637D+05	0.2799626D+04
44	0.8340500D+06	0.7344819D+06	0.9956806D+05
45	0.3720100D+05	0.3567887D+05	0.1522133D+04
46	0.8879500D+05	0.9051852D+05	-0.1723521D+04
47	0.1601100D+06	0.1619257D+06	-0.1815728D+04
48	0.1146300D+06	0.1053250D+06	0.9304971D+04
49	0.8406300D+05	0.7084466D+05	0.1321834D+05
50	0.2168200D+05	0.2168441D+05	-0.2406544D+01
51	0.5095900D+05	0.4953258D+05	0.1426415D+04
52	0.5377700D+05	0.4253678D+05	0.1124022D+05
53	0.6716700D+05	0.6733544D+05	-0.1684433D+03
54	0.1529300D+05	0.1075404D+05	0.4538961D+04
55	0.3626200D+05	0.2400937D+05	0.1225263D+05
56	0.1222400D+05	0.8228885D+04	0.3995115D+04
57	0.9907000D+04	0.6669986D+04	0.3237014D+04
58	0.2310300D+05	0.1349356D+05	0.9609440D+04
59	0.2278700D+05	0.1674567D+05	0.6041331D+04
60	0.3585400D+05	0.2136796D+05	0.1448604D+05
61	0.3893600D+06	0.4442260D+06	-0.5486602D+05
62	0.4266300D+05	0.4209914D+05	0.5638596D+03
63	0.7401600D+05	0.7765977D+05	-0.3643774D+04
64	0.4274600D+05	0.3994097D+05	0.2805029D+04
65	0.3194800D+05	0.2945606D+05	0.2491942D+04
66	0.8775000D+04	0.7526363D+04	0.1248637D+04
67	0.1715600D+05	0.1619422D+05	0.9617814D+03
68	0.1195300D+05	0.1020883D+05	0.1744171D+04
69	0.2736000D+07	0.5655396D+07	-0.2919396D+07
70	0.1289300D+07	0.1871782D+07	-0.5824821D+06
71	0.2139400D+06	0.2178555D+06	-0.3915529D+04
72	0.5161200D+06	0.4546783D+06	0.6144168D+05
73	0.1494000D+06	0.1257645D+06	0.2363554D+05
74	0.2677600D+06	0.2307226D+06	0.3703735D+05
75	0.8032200D+05	0.4301013D+05	0.3731187D+05
76	0.1010400D+06	0.9367755D+05	0.7362455D+04
77	0.2942800D+05	0.3876441D+05	-0.9336413D+04
78	0.2226600D+05	0.3048461D+05	-0.8218609D+04
79	0.7673200D+05	0.5938219D+05	0.1734981D+05
80	0.6434800D+05	0.5817662D+05	0.6171382D+04
81	0.1222000D+06	0.8454028D+05	0.3765972D+05
82	0.6370900D+06	0.1126867D+07	-0.4897768D+06
83	0.7216000D+05	0.7504575D+05	-0.2885752D+04
84	0.1315500D+06	0.1493928D+06	-0.1784282D+05
85	0.2025800D+05	0.1610669D+05	0.4151314D+04
86	0.3354000D+05	0.3156910D+05	0.1970898D+04
87	0.5720000D+04	0.1304642D+05	-0.7326421D+04
88	0.3154000D+04	0.8733989D+04	-0.5579989D+04
89	0.5015000D+04	0.1250024D+05	-0.7485236D+04
90	0.7885000D+06	0.1400739D+07	-0.6122388D+06
91	0.9269500D+05	0.1017410D+06	-0.9046037D+04
92	0.1692700D+06	0.1813698D+06	-0.1209983D+05
93	0.3411500D+05	0.2790925D+05	0.6205754D+04
94	0.5428600D+05	0.5318598D+05	0.1100017D+04
95	0.1421600D+05	0.2206886D+05	-0.7852859D+04
96	0.8656000D+04	0.1408496D+05	-0.5428958D+04
97	0.1496700D+05	0.2000469D+05	-0.5037688D+04
98	0.3195000D+06	0.4518501D+06	-0.1323501D+06
99	0.4823100D+05	0.4525536D+05	0.2975637D+04

```
 100  0.9668800D+05  0.9386816D+05  0.2819845D+04
 101  0.1993900D+05  0.1559567D+05  0.4343325D+04
 102  0.4677000D+04  0.8046066D+04 -0.3369066D+04
 103  0.2577000D+04  0.6491090D+04 -0.3914090D+04
 104  0.4218000D+04  0.9085370D+04 -0.4867370D+04
 105  0.1490900D+06  0.2114886D+06 -0.6239863D+05
 106  0.2599500D+06  0.3475991D+06 -0.8764910D+05
 107  0.1497300D+05  0.1243154D+05  0.2541460D+04
 108  0.2634000D+05  0.2431636D+05  0.2023639D+04
 109  0.9546000D+04  0.1465854D+05 -0.5112541D+04
 110  0.6170000D+04  0.8598725D+04 -0.2428725D+04
 111  0.9086000D+04  0.1364629D+05 -0.4560293D+04
 112  0.2581700D+06  0.5190232D+06 -0.2608532D+06
 113  0.7723000D+04  0.7697209D+04  0.2579056D+02
 114  0.9061000D+04  0.7886434D+04  0.1174566D+04
 115  0.1740200D+05  0.1742530D+05 -0.2330187D+02
 116  0.7988000D+04  0.1987733D+05 -0.1188933D+05
 117  0.3310000D+04  0.9004813D+04 -0.5694813D+04
 118  0.6790000D+04  0.1298105D+05 -0.6191046D+04
 119  0.1126800D+06  0.1316980D+06 -0.1901803D+05
 120  0.1116000D+05  0.6668290D+04  0.4491710D+04


CPU TIME:      0.02 sec
```

Output file: variant2.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         683,
 NAMY    =      'SHIP',
 NVAXZ   =           8,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           0,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = LOG+AU   ,
 NVARIANT        =           2
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
 PI      =   1.00000000000000     ,  0.000000000000000E+000,
 NTPI    = 2*0,
 MATFILE = R400miles.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           0,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           2,           3,            4,            5, 46*0,
 NUMLX   = 2*1,           0,            1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 10*0
 /
 &YXZDATA
 YXZFILE = America.in                                                                      ,
 YXZFMT  = (5e15.8)
```

```
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
SHIP        0.8749967D+01  0.3598531D+06  0.2035627D+08     0     0.1000000D-06
INDEPENDENT
AGDPM       0.1479850D+01  0.8925339D+02  0.6423090D+03     0     0.1000000D-01
AGDPX       0.1479850D+01  0.8938075D+02  0.6423090D+03     0     0.1000000D-01
CDUMMY      0.0000000D+00  0.1317716D+00  0.1000000D+01   593     0.1000000D+01
DIST        0.7200000D+02  0.1347331D+04  0.3448000D+04     0     0.1000000D-02
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1133D+01  0.2599D+03 -0.8885D-01  0.9672D+03

AGDPX        0.8823D+00  0.1000D+01  0.2293D+03 -0.7839D-01  0.8534D+03

CDUMMY       0.3848D-02  0.4361D-02  0.1000D+01 -0.3419D-03  0.3722D+01

DIST        -0.1126D+02 -0.1276D+02 -0.2925D+04  0.1000D+01 -0.1089D+05

CONSTANT     0.1034D-02  0.1172D-02  0.2687D+00 -0.9186D-04  0.1000D+01


INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


AGDPM         X1  0.1062377D+01  0.3083117D-01    34.4578896 -0.4496678D-09  0.4283308D+04
1.0623771
                                                                              0.1252196D+05
1.0623771

AGDPX         X1  0.1205750D+01  0.3113321D-01    38.7287443 -0.4228258D-09  0.4854433D+04
1.2057500
                                                                              0.1209211D+05
1.2057500

CDUMMY        X0  0.3093290D+01  0.1300891D+00    23.7782348 -0.1579625D-10  0.1113130D+07
3.0932899
                                                                              0.2906163D+07
3.0932899

DIST          X1 -0.1424849D+01  0.6264168D-01   -22.7460148 -0.9253610D-09 -0.3805570D+03     -
1.4248486
                                                                             -0.3913788D+04     -
1.4248486

CONSTANT      X0  0.1151202D+02  0.4875428D+00    23.6123263 -0.1061904D-09  0.4142637D+07
11.5120204
                                                                              0.1081561D+08
11.5120204

VAR(W)            0.1221816D+01  0.6611660D-01    18.4797186  0.0000000D+00

INITIAL VALUES

RHO1           0.0000000D+00
PI1            0.1000000D+01
```

```
LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00


LOG-LIKELIHOOD = -0.8089564D+04

NUMBER OF OBSERVATIONS             =           683

PEARSON-R2(E) (unadjusted)         =        0.5137
PEARSON-R2(E) (adjusted for D.F.)  =        0.5108

PSEUDO-R2(L)  (unadjusted)         =        0.8121
PSEUDO-R2(L)  (adjusted for D.F.)  =        0.8110

PROBABILITY of Y to be at the limit =       0.0000
MEAN VALUE of E(Y)                 =  0.9395055D+06

MEAN of residual W                 = -0.1900732D-12
VARIANCE of residual W             =  0.1221816D+01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.2166502D+01   Upper Bound =  0.2166502D+01



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         1      2000   0.1E+01   0.1E-06
  0.00000E+00
  0.10000E+01
  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
   0.8089563515D+04  0.00000D+00

new absolute minimum , parameters for fcn entry     6  follow - - f,x(i)
   0.8019243433D+04  0.58946D+00

step to unallowed region , parameters for fcn entry    11 follow - - x(i)
                     0.12248D+06

step to unallowed region , parameters for fcn entry    12 follow - - x(i)
                     0.61240D+05

step to unallowed region , parameters for fcn entry    13 follow - - x(i)
                     0.30620D+05

step to unallowed region , parameters for fcn entry    14 follow - - x(i)
                     0.15311D+05


the run has been terminated since no improvement can be found on the current minimum,
 the parameters of which follow   - - f,x(i)
   0.8019243433D+04  0.58946D+00


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was   0.8019243433D+04 for entry   26 - - x(i) follows
  0.58946E+00



run statistics follow - -

 nfcn=   28 nst=    4
```

```
MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1139D+01  0.2453D+03 -0.8911D-01  0.9524D+03

AGDPX        0.8782D+00  0.1000D+01  0.2154D+03 -0.7826D-01  0.8364D+03

CDUMMY       0.4077D-02  0.4642D-02  0.1000D+01 -0.3633D-03  0.3882D+01

DIST        -0.1122D+02 -0.1278D+02 -0.2753D+04  0.1000D+01 -0.1069D+05

CONSTANT     0.1050D-02  0.1196D-02  0.2576D+00 -0.9356D-04  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

AGDPM         X1  0.1084188D+01  0.4094374D-01     26.4799481                 0.8156823D-10
0.4371246D+04      1.0841882
                                 0.4094374D-01     26.4799481
0.9872440D+04      1.0841882

AGDPX         X1  0.1236296D+01  0.4362348D-01     28.3401565                 0.7964196D-10
0.4977415D+04      1.2362963
                                 0.4362348D-01     28.3401565
0.1027046D+05      1.2362963

CDUMMY        X0  0.2979836D+01  0.1261927D+00     23.6133789                 0.2213992D-11
0.1072304D+07      2.9798365
                                 0.1261927D+00     23.6133789
0.2006258D+07      2.9798365

DIST          X1 -0.1458389D+01  0.1057113D+00    -13.7959646                 0.1701661D-09 -
0.3895153D+03     -1.4583894
                                 0.1057113D+00    -13.7959646                               -
0.2781379D+04     -1.4583894

CONSTANT      X0  0.1156901D+02  0.7943373D+00     14.5643551                 0.3205935D-10
0.4163145D+07     11.5690105
                                 0.7943373D+00     14.5643551
0.7789161D+07     11.5690105

VAR(W)            0.9256905D+00  0.4313337D-01     21.4611193                 0.0000000D+00
                                 0.4313337D-01     21.4611193

RHO1      (E)     0.5295103D+00  0.4866401D-01     10.8809418                 0.1207480D-04
                                 0.4866401D-01     10.8809418
PI1       (F)     0.1000000D+01

LAM(Y)    (F)     0.0000000D+00
LAM(X1)   (F)     0.0000000D+00


LOG-LIKELIHOOD = -0.8019243D+04

NUMBER OF OBSERVATIONS            =            683

PEARSON-R2(E) (unadjusted)        =         0.5815
PEARSON-R2(E) (adjusted for D.F.) =         0.5784
```

```
PSEUDO-R2(L)  (unadjusted)          =         0.7791
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.7775

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.6732780D+06

MEAN of residual W                  =  0.3867546D-02
VARIANCE of residual W              =  0.9256905D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.1881904D+01   Upper Bound =  0.1889639D+01


CPU TIME:  1 min 59 sec
```

Output file: variant3.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         683,
 NAMY    =      'SHIP',
 NVAXZ   =           8,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           1,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = LOG+AU+PR,
 NVARIANT        =           3
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
 PI      =  0.990000000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R400miles.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           0,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           2,           3,           4,           5, 46*0,
 NUMLX   = 2*1,           0,           1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 10*0
 /
 &YXZDATA
 YXZFILE = America.in                                                                   ,
 YXZFMT  = (5e15.8)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
SHIP        0.8749967D+01  0.3598531D+06  0.2035627D+08     0     0.1000000D-06
INDEPENDENT
AGDPM       0.1479850D+01  0.8925339D+02  0.6423090D+03     0     0.1000000D-01
AGDPX       0.1479850D+01  0.8938075D+02  0.6423090D+03     0     0.1000000D-01
CDUMMY      0.0000000D+00  0.1317716D+00  0.1000000D+01   593     0.1000000D+01
DIST        0.7200000D+02  0.1347331D+04  0.3448000D+04     0     0.1000000D-02
```

```
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01

MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1133D+01  0.2599D+03 -0.8885D-01 -0.3363D+03

AGDPX        0.8823D+00  0.1000D+01  0.2293D+03 -0.7839D-01 -0.2968D+03

CDUMMY       0.3848D-02  0.4361D-02  0.1000D+01 -0.3419D-03 -0.1294D+01

DIST        -0.1126D+02 -0.1276D+02 -0.2925D+04  0.1000D+01  0.3786D+04

CONSTANT    -0.2973D-02 -0.3370D-02 -0.7727D+00  0.2642D-03  0.1000D+01


INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


AGDPM         X1  0.1062377D+01  0.3083117D-01     34.4578896 -0.1044276D-10  0.4283308D+04
1.0623771
                                                                              0.1252196D+00
1.0623771

AGDPX         X1  0.1205750D+01  0.3113321D-01     38.7287443 -0.6419000D-11  0.4854433D+04
1.2057500
                                                                              0.1209211D+00
1.2057500

CDUMMY        X0  0.3093290D+01  0.1300891D+00     23.7782348 -0.2907732D-12  0.1113130D+07
3.0932899
                                                                              0.2906163D+00
3.0932899

DIST          X1 -0.1424849D+01  0.6264168D-01    -22.7460148 -0.8677762D-14 -0.3805570D+03     -
1.4248486
                                                                             -0.3913788D+00     -
1.4248486

CONSTANT      X0 -0.4003469D+01  0.6032814D-01    -66.3615579 -0.1757724D-11 -0.1440661D+07     -
4.0034694
                                                                             -0.3761282D+00     -
4.0034694

VAR(W)            0.1221816D+01  0.6611660D-01     18.4797186  0.0000000D+00

INITIAL VALUES

RHO1           0.0000000D+00
PI1            0.9900000D+00

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00


LOG-LIKELIHOOD =  0.2919096D+04

NUMBER OF OBSERVATIONS             =            683

PEARSON-R2(E) (unadjusted)         =         0.5137
```

```
PEARSON-R2(E) (adjusted for D.F.)   =         0.5108

PSEUDO-R2(L)  (unadjusted)          =         0.8121
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.8110

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.9395055D-01

MEAN of residual W                  = -0.2848199D-14
VARIANCE of residual W              =  0.1221816D+01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.2166502D+01   Upper Bound =  0.2166502D+01



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         2      2000   0.1E+01   0.1E-06
  0.00000E+00  0.45951E+01
  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.2919095814D+04  0.00000D+00  0.45951D+01

new absolute minimum , parameters for fcn entry     6  follow - - f,x(i)
  -0.2989636137D+04  0.59291D+00  0.45951D+01

new absolute minimum , parameters for fcn entry    18  follow - - f,x(i)
  -0.2993863560D+04  0.59693D+00  0.27798D+00

new absolute minimum , parameters for fcn entry    21  follow - - f,x(i)
  -0.2993872014D+04  0.59749D+00  0.20062D+00

new absolute minimum , parameters for fcn entry    25  follow - - f,x(i)
  -0.2996963890D+04  0.77917D+00 -0.63671D+00

new absolute minimum , parameters for fcn entry    28  follow - - f,x(i)
  -0.2997108787D+04  0.78588D+00 -0.43741D+00

new absolute minimum , parameters for fcn entry    30  follow - - f,x(i)
  -0.2997116656D+04  0.79631D+00 -0.47311D+00

new absolute minimum , parameters for fcn entry    32  follow - - f,x(i)
  -0.2997116666D+04  0.79626D+00 -0.47475D+00

new absolute minimum , parameters for fcn entry    33  follow - - f,x(i)
  -0.2997116666D+04  0.79625D+00 -0.47474D+00

new absolute minimum , parameters for fcn entry    34  follow - - f,x(i)
  -0.2997116666D+04  0.79625D+00 -0.47474D+00


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.2997116666D+04  0.79625D+00 -0.47474D+00


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.2997116666D+04 for entry   34 - - x(i) follows
  0.79625E+00 -0.47474E+00
```

```
run statistics follow - -

 nfcn=   38 nst=   10

MRS of X(l) with respect to X(k) - at the sample means
==================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
==================================================================================================
==============================

AGDPM        0.1000D+01  0.1160D+01  0.2557D+03 -0.9629D-01  0.1034D+04

AGDPX        0.8618D+00  0.1000D+01  0.2204D+03 -0.8298D-01  0.8907D+03

CDUMMY       0.3911D-02  0.4538D-02  0.1000D+01 -0.3765D-03  0.4042D+01

DIST        -0.1039D+02 -0.1205D+02 -0.2656D+04  0.1000D+01 -0.1073D+05

CONSTANT     0.9675D-03  0.1123D-02  0.2474D+00 -0.9316D-04  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

AGDPM         X1  0.1055374D+01  0.4126054D-01     25.5782999                 -0.1598546D-10
0.4255074D+04      1.0553745
                                 0.4126054D-01     25.5782999
0.9977027D+04      1.0553745

AGDPX         X1  0.1226404D+01  0.4310285D-01     28.4529723                 -0.5265199D-11
0.4937588D+04      1.2264041
                                 0.4310285D-01     28.4529723
0.1059039D+05      1.2264041

CDUMMY        X0  0.3023654D+01  0.1256762D+00     24.0590809                 -0.9105838D-12
0.1088071D+07      3.0236542
                                 0.1256762D+00     24.0590809
0.2029716D+07      3.0236542

DIST          X1 -0.1533994D+01  0.1115994D+00    -13.7455451                 -0.5045893D-11 -
0.4097083D+03     -1.5339942
                                 0.1115994D+00    -13.7455451                                  -
0.2916469D+04     -1.5339942

CONSTANT      X0  0.1222146D+02  0.8310139D+00     14.7066806                  0.5560139D-11
0.4397929D+07     12.2214561
                                 0.8310139D+00     14.7066806
0.8204007D+07     12.2214561

VAR(W)            0.8220137D+00  0.4217483D-01     19.4906236                  0.0000000D+00
                                 0.4217483D-01     19.4906236

RHO1      (E)     0.6619374D+00  0.5773565D-01     11.4649688                  0.1625423D-07
                                 0.5773565D-01     11.4649688
PI1       (E)     0.3834955D+00  0.1227180D+00      3.1250152     -5.0237504  0.5663203D-10
                                 0.1227180D+00      3.1250152     -5.0237504

LAM(Y)    (F)     0.0000000D+00
LAM(X1)   (F)     0.0000000D+00


LOG-LIKELIHOOD = -0.8011543D+04
```

```
NUMBER OF OBSERVATIONS              =           683

PEARSON-R2(E) (unadjusted)          =        0.5754
PEARSON-R2(E) (adjusted for D.F.)   =        0.5716

PSEUDO-R2(L)  (unadjusted)          =        0.7937
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.7919

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.6712790D+06

MEAN of residual W                  =  0.7784373D-02
VARIANCE of residual W              =  0.8220137D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.1769249D+01   Upper Bound =  0.1784818D+01


CPU TIME: 9 min 54 sec
```

Output file: variant4.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         683,
 NAMY    =      'SHIP',
 NVAXZ   =           8,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           1,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = BC1+AU+PR,
 NVARIANT        =           4
 /
 &AUTOCOR
 RHO     =  0.661937400000000     ,  0.000000000000000E+000,
 PI      =  0.383496000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R400miles.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           1,
 NUMLXY  =           1
 /
 &XBOXCOX
 NUMX    =           2,           3,           4,           5, 46*0,
 NUMLX   = 2*1,           0,           1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    =           1, 9*0
 /
 &YXZDATA
 YXZFILE = America.in                                                                      ,
 YXZFMT  = (5e15.8)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
SHIP        0.8749967D+01  0.3598531D+06  0.2035627D+08     0     0.1000000D-06
```

```
INDEPENDENT
AGDPM       0.1479850D+01  0.8925339D+02  0.6423090D+03     0     0.1000000D-01
AGDPX       0.1479850D+01  0.8938075D+02  0.6423090D+03     0     0.1000000D-01
CDUMMY      0.0000000D+00  0.1317716D+00  0.1000000D+01   593     0.1000000D+01
DIST        0.7200000D+02  0.1347331D+04  0.3448000D+04     0     0.1000000D-02
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01

MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
       X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1160D+01  0.2557D+03 -0.9629D-01 -0.3370D+03

AGDPX        0.8618D+00  0.1000D+01  0.2204D+03 -0.8298D-01 -0.2904D+03

CDUMMY       0.3911D-02  0.4538D-02  0.1000D+01 -0.3765D-03 -0.1318D+01

DIST        -0.1039D+02 -0.1205D+02 -0.2656D+04  0.1000D+01  0.3500D+04

CONSTANT    -0.2967D-02 -0.3443D-02 -0.7587D+00  0.2857D-03  0.1000D+01


INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


AGDPM         X1  0.1055375D+01  0.4161015D-01     25.3633903 -0.6715252D-12  0.4255075D+04
1.0553745
                                                                              0.9977027D-01
1.0553745

AGDPX         X1  0.1226404D+01  0.4160387D-01     29.4781291  0.7871377D-12  0.4937589D+04
1.2264041
                                                                              0.1059039D+00
1.2264041

CDUMMY        X0  0.3023654D+01  0.1147141D+00     26.3581647 -0.1514038D-12  0.1088071D+07
3.0236541
                                                                              0.2029715D+00
3.0236541

DIST          X1 -0.1533994D+01  0.1018111D+00    -15.0670567  0.1348588D-12 -0.4097083D+03     -
1.5339942
                                                                             -0.2916469D+00     -
1.5339942

CONSTANT      X0 -0.3985118D+01  0.1146969D+00    -34.7447580 -0.2901254D-11 -0.1434057D+07     -
3.9851176
                                                                             -0.2675126D+00     -
3.9851176

VAR(W)            0.8220137D+00  0.4448194D-01     18.4797186  0.0000000D+00

INITIAL VALUES

RHO1           0.6619374D+00
PI1            0.3834960D+00

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00

LAM(Y) is set equal to LAM(X1).
```

```
LOG-LIKELIHOOD =  0.2997117D+04

NUMBER OF OBSERVATIONS              =           683

PEARSON-R2(E) (unadjusted)          =        0.5754
PEARSON-R2(E) (adjusted for D.F.)   =        0.5729

PSEUDO-R2(L)  (unadjusted)          =        0.7662
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.7648

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.6712790D-01

MEAN of residual W                  =  0.7784372D-02
VARIANCE of residual W              =  0.8220137D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.1769249D+01   Upper Bound =  0.1784818D+01



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         3      2000   0.1E+01   0.1E-06
  0.79625E+00 -0.47474E+00  0.00000E+00
  0.10000E+01  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.2997116666D+04  0.79625D+00 -0.47474D+00  0.00000D+00

new absolute minimum , parameters for fcn entry     5  follow - - f,x(i)
  -0.3032275300D+04  0.79625D+00 -0.47474D+00  0.63929D-01

new absolute minimum , parameters for fcn entry     7  follow - - f,x(i)
  -0.3032301712D+04  0.81279D+00 -0.47296D+00  0.63978D-01

new absolute minimum , parameters for fcn entry     9  follow - - f,x(i)
  -0.3032302084D+04  0.81142D+00 -0.45796D+00  0.63965D-01

new absolute minimum , parameters for fcn entry    11  follow - - f,x(i)
  -0.3032302094D+04  0.81140D+00 -0.45806D+00  0.63931D-01

new absolute minimum , parameters for fcn entry    12  follow - - f,x(i)
  -0.3032302094D+04  0.81140D+00 -0.45806D+00  0.63931D-01

new absolute minimum , parameters for fcn entry    13  follow - - f,x(i)
  -0.3032302094D+04  0.81140D+00 -0.45806D+00  0.63931D-01

new absolute minimum , parameters for fcn entry    14  follow - - f,x(i)
  -0.3032302094D+04  0.81140D+00 -0.45806D+00  0.63931D-01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3032302094D+04  0.81140D+00 -0.45806D+00  0.63931D-01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3032302094D+04 for entry   14 - - x(i) follows
  0.81140E+00 -0.45806E+00  0.63931E-01
```

```
run statistics follow - -

 nfcn=   14 nst=    7

MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM         0.1000D+01  0.1125D+01  0.2457D+03 -0.1006D+00  0.7391D+03

AGDPX         0.8887D+00  0.1000D+01  0.2183D+03 -0.8940D-01  0.6568D+03

CDUMMY        0.4070D-02  0.4580D-02  0.1000D+01 -0.4095D-03  0.3008D+01

DIST         -0.9940D+01 -0.1119D+02 -0.2442D+04  0.1000D+01 -0.7347D+04

CONSTANT      0.1353D-02  0.1523D-02  0.3324D+00 -0.1361D-03  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

AGDPM         X1  0.1618806D+01  0.1054975D+00     15.3444932                  0.6078623D-11
0.3838761D+04      0.9521174
                                 0.6048706D-01     26.7628551
0.5432650D+04      1.0169236

AGDPX         X1  0.1824071D+01  0.1061278D+00     17.1874983                  0.8058052D-11
0.4319747D+04      1.0729440
                                 0.6248188D-01     29.1936074
0.5625180D+04      1.1453676

CDUMMY        X0  0.5938269D+01  0.5437883D+00     10.9201862                  0.4720201D-13
0.9431370D+06      2.6208942
                                 0.2095469D+00     28.3386232
0.8478418D+06      2.7080416

DIST          X1 -0.2066684D+01  0.1577009D+00    -13.1050868                  0.2065168D-10 -
0.3861767D+03     -1.4458893
                                 0.1362860D+00    -15.1643159                                -
0.1065894D+04     -1.6127310

CONSTANT      X0  0.1786423D+02  0.1589638D+01     11.2379262                  0.6004208D-11
0.2837261D+07      7.8844956
                                 0.1299595D+01     13.7459958
0.2550582D+07      8.8679503

VAR(W)            0.2766031D+01  0.4650395D+00      5.9479488                  0.0000000D+00
                                 0.1490444D+00     18.5584421

RHO1      (E)     0.6703606D+00  0.5995131D-01     11.1817503                 -0.3191377D-07
                                 0.5992258D-01     11.1871127
PI1       (E)     0.3874453D+00  0.1462979D+00      2.6483309     -4.1870358 -0.8123020D-08
                                 0.1462683D+00      2.6488677     -4.1878844

LAM(Y)    (E)     0.6393131D-01  0.7663071D-02      8.3427799   -122.1532061  0.1344233D-07
LAM(X1)   (E)     0.6393131D-01  0.7663071D-02      8.3427799   -122.1532061  0.1344233D-07

LAM(Y) is set equal to LAM(X1).
```

```
LOG-LIKELIHOOD = -0.7976357D+04

NUMBER OF OBSERVATIONS              =           683

PEARSON-R2(E) (unadjusted)          =        0.6839
PEARSON-R2(E) (adjusted for D.F.)   =        0.6806

PSEUDO-R2(L)  (unadjusted)          =        0.7990
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.7969

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.3621477D+06

MEAN of residual W                  =  0.1448522D-01
VARIANCE of residual W              =  0.2766031D+01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.3245267D+01   Upper Bound =  0.3274238D+01


CPU TIME:  3 min 36 sec
```

Output file: variant5.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         683,
 NAMY    =      ‘SHIP’,
 NVAXZ   =           8,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           1,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = BC2+AU+PR,
 NVARIANT        =           5
 /
 &AUTOCOR
 RHO     =  0.661937400000000     ,  0.000000000000000E+000,
 PI      =  0.383496000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R400miles.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           1,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           2,           3,           4,           5, 46*0,
 NUMLX   = 2*1,           0,           1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    =           1, 9*0
 /
 &YXZDATA
 YXZFILE = America.in                                                                        ,
 YXZFMT  = (5e15.8)
 /
```

```
VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
SHIP        0.8749967D+01  0.3598531D+06  0.2035627D+08     0     0.1000000D-06
INDEPENDENT
AGDPM       0.1479850D+01  0.8925339D+02  0.6423090D+03     0     0.1000000D-01
AGDPX       0.1479850D+01  0.8938075D+02  0.6423090D+03     0     0.1000000D-01
CDUMMY      0.0000000D+00  0.1317716D+00  0.1000000D+01   593     0.1000000D+01
DIST        0.7200000D+02  0.1347331D+04  0.3448000D+04     0     0.1000000D-02
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1160D+01  0.2557D+03 -0.9629D-01 -0.3370D+03

AGDPX        0.8618D+00  0.1000D+01  0.2204D+03 -0.8298D-01 -0.2904D+03

CDUMMY       0.3911D-02  0.4538D-02  0.1000D+01 -0.3765D-03 -0.1318D+01

DIST        -0.1039D+02 -0.1205D+02 -0.2656D+04  0.1000D+01  0.3500D+04

CONSTANT    -0.2967D-02 -0.3443D-02 -0.7587D+00  0.2857D-03  0.1000D+01


INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


AGDPM         X1  0.1055375D+01  0.4161015D-01     25.3633903 -0.6715252D-12  0.4255075D+04
1.0553745
                                                                              0.9977027D-01
1.0553745

AGDPX         X1  0.1226404D+01  0.4160387D-01     29.4781291  0.7871377D-12  0.4937589D+04
1.2264041
                                                                              0.1059039D+00
1.2264041

CDUMMY        X0  0.3023654D+01  0.1147141D+00     26.3581647 -0.1514038D-12  0.1088071D+07
3.0236541
                                                                              0.2029715D+00
3.0236541

DIST          X1 -0.1533994D+01  0.1018111D+00    -15.0670567  0.1348588D-12 -0.4097083D+03     -
1.5339942
                                                                             -0.2916469D+00     -
1.5339942

CONSTANT      X0 -0.3985118D+01  0.1146969D+00    -34.7447580 -0.2901254D-11 -0.1434057D+07     -
3.9851176
                                                                             -0.2675126D+00     -
3.9851176

VAR(W)            0.8220137D+00  0.4448194D-01     18.4797186  0.0000000D+00

INITIAL VALUES

RHO1           0.6619374D+00
PI1            0.3834960D+00

LAM(Y)         0.0000000D+00
```

```
LAM(X1)        0.0000000D+00


LOG-LIKELIHOOD =  0.2997117D+04

NUMBER OF OBSERVATIONS              =            683

PEARSON-R2(E) (unadjusted)          =         0.5754
PEARSON-R2(E) (adjusted for D.F.)   =         0.5729

PSEUDO-R2(L)  (unadjusted)          =         0.7662
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.7648

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.6712790D-01

MEAN of residual W                  =  0.7784372D-02
VARIANCE of residual W              =  0.8220137D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.1769249D+01   Upper Bound =  0.1784818D+01



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         4      2000   0.1E+01   0.1E-06
  0.79625E+00 -0.47474E+00  0.00000E+00  0.00000E+00
  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.2997116666D+04  0.79625D+00 -0.47474D+00  0.00000D+00  0.00000D+00

new absolute minimum , parameters for fcn entry     6  follow - - f,x(i)
  -0.3032142612D+04  0.79625D+00 -0.47474D+00  0.64131D-01  0.40989D-03

new absolute minimum , parameters for fcn entry     9  follow - - f,x(i)
  -0.3033196608D+04  0.79393D+00 -0.47443D+00  0.65901D-01  0.33605D-01

new absolute minimum , parameters for fcn entry    11  follow - - f,x(i)
  -0.3033199205D+04  0.78807D+00 -0.46892D+00  0.65907D-01  0.33225D-01

new absolute minimum , parameters for fcn entry    13  follow - - f,x(i)
  -0.3033234784D+04  0.77645D+00 -0.31515D+00  0.65944D-01  0.32137D-01

new absolute minimum , parameters for fcn entry    15  follow - - f,x(i)
  -0.3033234869D+04  0.77633D+00 -0.31506D+00  0.65851D-01  0.32184D-01

new absolute minimum , parameters for fcn entry    16  follow - - f,x(i)
  -0.3033234869D+04  0.77634D+00 -0.31509D+00  0.65852D-01  0.32189D-01

new absolute minimum , parameters for fcn entry    17  follow - - f,x(i)
  -0.3033234869D+04  0.77634D+00 -0.31510D+00  0.65852D-01  0.32190D-01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3033234869D+04  0.77634D+00 -0.31510D+00  0.65852D-01  0.32190D-01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3033234869D+04 for entry   17 - - x(i) follows
  0.77634E+00 -0.31510E+00  0.65852E-01  0.32190E-01
```

```
run statistics follow - -

 nfcn=   21 nst=    8

MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1127D+01  0.2526D+03 -0.9947D-01  0.7815D+03

AGDPX        0.8872D+00  0.1000D+01  0.2241D+03 -0.8825D-01  0.6934D+03

CDUMMY       0.3959D-02  0.4462D-02  0.1000D+01 -0.3938D-03  0.3094D+01

DIST        -0.1005D+02 -0.1133D+02 -0.2540D+04  0.1000D+01 -0.7857D+04

CONSTANT     0.1280D-02  0.1442D-02  0.3232D+00 -0.1273D-03  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

AGDPM         X1  0.1858078D+01  0.2128678D+00      8.7287888                 0.1500312D-10
0.3727988D+04      0.9246427
                                 0.6911426D-01     26.8841498
0.5452348D+04      1.0161081

AGDPX         X1  0.2097119D+01  0.2278715D+00      9.2030773                 0.1807309D-10
0.4201789D+04      1.0436454
                                 0.7163953D-01     29.2732067
0.5624687D+04      1.1465617

CDUMMY        X0  0.6076821D+01  0.5619138D+00     10.8145087                 0.2864961D-12
0.9417208D+06      2.6169585
                                 0.2150699D+00     28.2551040
0.8313837D+06      2.7089380

DIST          X1 -0.2556556D+01  0.4365902D+00     -5.8557345                 0.3218582D-10 -
0.3708199D+03     -1.3883915
                                 0.1664827D+00    -15.3562847                               -
0.1053400D+04     -1.5609076

CONSTANT      X0  0.1880105D+02  0.1899040D+01      9.9002911                -0.5885759D-12
0.2913586D+07      8.0965967
                                 0.1413203D+01     13.3038588
0.2572214D+07      9.1426409

VAR(W)            0.2892432D+01  0.4928048D+00      5.8693268                 0.0000000D+00
                                 0.1587858D+00     18.2159376

RHO1      (E)     0.6505984D+00  0.6249835D-01     10.4098493                 0.2035899D-05
                                 0.6220657D-01     10.4586765
PI1       (E)     0.4218712D+00  0.1592772D+00      2.6486598    -3.6297020 -0.1388720D-05
                                 0.1592615D+00      2.6489205    -3.6300593

LAM(Y)    (E)     0.6585158D-01  0.7726522D-02      8.5227968  -120.9015385  0.2739936D-05
LAM(X1)   (E)     0.3218950D-01  0.2379766D-01      1.3526330   -40.6683025  0.9120507D-05
```

```
LOG-LIKELIHOOD = -0.7975424D+04

NUMBER OF OBSERVATIONS              =           683

PEARSON-R2(E) (unadjusted)          =        0.6828
PEARSON-R2(E) (adjusted for D.F.)   =        0.6790

PSEUDO-R2(L)  (unadjusted)          =        0.7986
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.7962

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.3559349D+06

MEAN of residual W                  =  0.1528882D-01
VARIANCE of residual W              =  0.2892432D+01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.3318113D+01   Upper Bound =  0.3348691D+01


CPU TIME:  4 min  18 sec
```

Output file: variant6.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         683,
 NAMY    =      'SHIP',
 NVAXZ   =           8,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           1,
 NLAMZ   =           1,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           1,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = BC2GHAUPR,
 NVARIANT        =           6
 /
 &AUTOCOR
 RHO     =  0.650000000000000     ,  0.000000000000000E+000,
 PI      =  0.421000000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R400miles.in

 /
 &HETERO
 NUMZ    =           8, 9*0,
 NUMLZ   =           1, 9*0,
 DELTA   = 10*0.000000000000000E+000  ,
 NTDEL   =           1, 9*0,
 VLZ     = 10*0.000000000000000E+000  ,
 NTLZ    = 10*0
 /
 &YBOXCOX
 VLY     =  6.580000000000000E-002,
 NTLY    =           1,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           2,           3,           4,           5, 46*0,
 NUMLX   = 2*1,           0,           1, 46*0,
 VLX     =  3.210000000000000E-002, 9*0.000000000000000E+000  ,
```

```
 NTLX    =           1, 9*0
 /
 &YXZDATA
 YXZFILE = America.in                                                                        ,
 YXZFMT  = (5e15.8)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
SHIP        0.8749967D+01  0.3598531D+06  0.2035627D+08     0     0.1000000D-06
INDEPENDENT
AGDPM       0.1479850D+01  0.8925339D+02  0.6423090D+03     0     0.1000000D-01
AGDPX       0.1479850D+01  0.8938075D+02  0.6423090D+03     0     0.1000000D-01
CDUMMY      0.0000000D+00  0.1317716D+00  0.1000000D+01   593     0.1000000D+01
DIST        0.7200000D+02  0.1347331D+04  0.3448000D+04     0     0.1000000D-02
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01
HETEROSKEDASTICITY
DPMDPX      0.9629014D+01  0.6419071D+04  0.1354784D+06     0     0.1000000D-04


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1127D+01  0.2179D+03 -0.1071D+00 -0.3603D+03

AGDPX        0.8872D+00  0.1000D+01  0.1933D+03 -0.9502D-01 -0.3197D+03

CDUMMY       0.4589D-02  0.5172D-02  0.1000D+01 -0.4914D-03 -0.1653D+01

DIST        -0.9337D+01 -0.1052D+02 -0.2035D+04  0.1000D+01  0.3365D+04

CONSTANT    -0.2775D-02 -0.3128D-02 -0.6048D+00  0.2972D-03  0.1000D+01


INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


AGDPM         X1  0.7456722D+00  0.2794662D-01     26.6820193  0.1205673D-11  0.1496479D+04
0.3711675
                                                                              0.5453880D-01
1.0160563

AGDPX         X1  0.8416289D+00  0.2792364D-01     30.1403724 -0.1516784D-11  0.1686723D+04
0.4189503
                                                                              0.5626016D-01
1.1465405

CDUMMY        X0  0.2103064D+01  0.7439030D-01     28.2706686 -0.3596514D-12  0.3261254D+06
0.9062737
                                                                              0.8317907D-01
2.7091855

DIST          X1 -0.1104908D+01  0.6664730D-01    -16.5784402  0.1784640D-12 -0.1602656D+03     -
0.6000526
                                                                             -0.1053971D+00     -
1.5606610

CONSTANT      X0 -0.3477384D+01  0.7244016D-01    -48.0035409 -0.7288589D-11 -0.5392435D+06     -
1.4985099
                                                                             -0.1375353D+00     -
4.8862779
```

```
VAR(W)            0.3465092D+00  0.1875078D-01     18.4797186  0.0000000D+00

INITIAL VALUES

RHO1           0.6500000D+00
PI1            0.4210000D+00

DPMDPX     Z1  0.0000000D+00
LAM(Z1)        0.0000000D+00

LAM(Y)         0.6580000D-01
LAM(X1)        0.3210000D-01


LOG-LIKELIHOOD =  0.3033235D+04

NUMBER OF OBSERVATIONS              =           683

PEARSON-R2(E) (unadjusted)          =        0.6827
PEARSON-R2(E) (adjusted for D.F.)   =        0.6808

PSEUDO-R2(L)  (unadjusted)          =        0.7864
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.7852

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.3560468D-01

MEAN of residual W                  =  0.5278887D-02
VARIANCE of residual W              =  0.3465092D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.1148476D+01   Upper Bound =  0.1159034D+01



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         5      2000   0.1E+01   0.1E-06
  0.77530E+00 -0.31867E+00  0.00000E+00  0.65800E-01  0.32100E-01
  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.3033234649D+04  0.77530D+00 -0.31867D+00  0.00000D+00  0.65800D-01  0.32100D-01

new absolute minimum , parameters for fcn entry     4  follow - - f,x(i)
  -0.3043429453D+04  0.77554D+00 -0.31864D+00 -0.13822D+00  0.66689D-01  0.32108D-01

new absolute minimum , parameters for fcn entry     7  follow - - f,x(i)
  -0.3047986351D+04  0.77580D+00 -0.31858D+00 -0.20028D+00  0.38886D-01  0.31961D-01

new absolute minimum , parameters for fcn entry     9  follow - - f,x(i)
  -0.3048259165D+04  0.77977D+00 -0.31803D+00 -0.20395D+00  0.37363D-01  0.13929D-01

new absolute minimum , parameters for fcn entry    11  follow - - f,x(i)
  -0.3048297170D+04  0.79945D+00 -0.31331D+00 -0.20446D+00  0.37235D-01  0.15727D-01

new absolute minimum , parameters for fcn entry    13  follow - - f,x(i)
  -0.3048343489D+04  0.78438D+00 -0.15227D+00 -0.20682D+00  0.37139D-01  0.14201D-01

new absolute minimum , parameters for fcn entry    15  follow - - f,x(i)
  -0.3048353714D+04  0.78510D+00 -0.16345D+00 -0.21147D+00  0.37058D-01  0.14165D-01

new absolute minimum , parameters for fcn entry    16  follow - - f,x(i)
  -0.3048353732D+04  0.78524D+00 -0.16469D+00 -0.21158D+00  0.37006D-01  0.14147D-01

new absolute minimum , parameters for fcn entry    17  follow - - f,x(i)
```

```
  -0.3048353732D+04  0.78524D+00 -0.16475D+00 -0.21158D+00  0.37006D-01  0.14147D-01

new absolute minimum , parameters for fcn entry    18  follow - - f,x(i)
  -0.3048353732D+04  0.78524D+00 -0.16475D+00 -0.21158D+00  0.37006D-01  0.14147D-01

new absolute minimum , parameters for fcn entry    21  follow - - f,x(i)
  -0.3048353732D+04  0.78524D+00 -0.16475D+00 -0.21158D+00  0.37006D-01  0.14147D-01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3048353732D+04  0.78524D+00 -0.16475D+00 -0.21158D+00  0.37006D-01  0.14147D-01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3048353732D+04 for entry   21 - - x(i) follows
  0.78524E+00 -0.16475E+00 -0.21158E+00  0.37006E-01  0.14147E-01



run statistics follow - -

 nfcn=   21 nst=   10


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) AGDPM       AGDPX       CDUMMY      DIST        CONSTANT
X(l)
====================================================================================================
==============================

AGDPM        0.1000D+01  0.1082D+01  0.2494D+03 -0.8428D-01  0.7515D+03

AGDPX        0.9241D+00  0.1000D+01  0.2305D+03 -0.7788D-01  0.6945D+03

CDUMMY       0.4009D-02  0.4338D-02  0.1000D+01 -0.3379D-03  0.3013D+01

DIST        -0.1187D+02 -0.1284D+02 -0.2960D+04  0.1000D+01 -0.8917D+04

CONSTANT     0.1331D-02  0.1440D-02  0.3319D+00 -0.1121D-03  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y        ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)     ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

AGDPM         X1  0.1535415D+01  0.2073716D+00      7.4041727                   0.2984724D-10
0.4108789D+04      1.0190918
                                 0.5946279D-01     25.8214484
0.6627495D+04      1.0732464

AGDPX         X1  0.1663845D+01  0.2179026D+00      7.6357307                   0.4160706D-10
0.4446215D+04      1.1043561
                                 0.5918711D-01     28.1116149
0.6724342D+04      1.1630584

CDUMMY        X0  0.4572403D+01  0.5176071D+00      8.8337326                   0.3320116D-12
0.1024853D+07      2.8479749
                                 0.1672563D+00     27.3377021
0.1017430D+07      2.8607160
```

```
DIST          X1 -0.1879798D+01  0.3809022D+00     -4.9351195                0.7170369D-10 -
0.3462797D+03     -1.2965103
                                 0.1262708D+00    -14.8870311                              -
0.1150327D+04     -1.3782023

CONSTANT      X0  0.1377610D+02  0.1665206D+01      8.2729075                0.7848783D-11
0.3087758D+07      8.5806056
                                 0.1081476D+01     12.7382423
0.3065394D+07      9.1417368

VAR(W)            0.7304477D+01  0.1702314D+01      4.2909102                0.0000000D+00
                                 0.1682822D+01      4.3406107

RHO1      (E)     0.6557043D+00  0.6254177D-01     10.4842615               -0.1450088D-05
                                 0.6197691D-01     10.5798158
PI1       (E)     0.4589063D+00  0.1561296D+00      2.9392658     -3.4656707  0.2884307D-07
                                 0.1556826D+00      2.9477039     -3.4756200

DPMDPX    (E) Z1 -0.2115773D+00  0.3714243D-01     -5.6963777               -0.2457773D-06 -
0.5191198D+01     -0.0926007
                                 0.2995543D-01     -7.0630706                              -
0.2710842D+02     -0.0854658
LAM(Z1)   (F)     0.0000000D+00

LAM(Y)    (E)     0.3700570D-01  0.9255417D-02      3.9982749   -104.0465578 -0.7754330D-05
LAM(X1)   (E)     0.1414714D-01  0.2712890D-01      0.5214785    -36.3395772  0.4635563D-06


LOG-LIKELIHOOD = -0.7960306D+04

NUMBER OF OBSERVATIONS              =           683

PEARSON-R2(E) (unadjusted)          =        0.6344
PEARSON-R2(E) (adjusted for D.F.)   =        0.6296

PSEUDO-R2(L)  (unadjusted)          =        0.9062
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9049

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.3843427D+06

MEAN of residual W                  =  0.2402695D-01
VARIANCE of residual W              =  0.7304477D+01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.5273225D+01   Upper Bound =  0.5321279D+01


CPU TIME:  5 min  8 sec
```

1.11 Second Example of Input Files

The second example can be found in directory *d:\Level2new\Level2Tests\Second Example*. It illustrates the estimation of the six variants reported in **Table B.2 Generation-Distribution Models, First Order AR-C-D Process Utility from Variant 3 (Table B.1), Canada 1976 (Symmetric Flows)** of *Gaudry et al.*(1994, 2008), p. 51. The corresponding batch files (.bat) are also stored in this directory.

The data file *Canada1976.in* contains 120 O-D pairs (observations). Since the whole file is too big to be reproduced here, only the first three observations are listed below. The first variable is the dependent variable TOTFLOW (passenger flows), followed by four independent variables (POP, REVE, LANGFR, UTILITY) whose definitions will be given later in **section 2.5: Second Example of Input File** for L-2.1 Tablex. The Fortran format for the observations is (5e15.5): it indicates that 5 values are entered on a single line and that the field for each value is 15 characters. Each value entered is right-justified and contains 5 decimals after the point followed by the exponential of the form E+XX or E-XX.

Data file: Canada1976.in

```
TOTFLOW
POP
REVE
LANGFR
UTILITY
    0.38651E+05    0.43104E+06    0.41887E+04    0.68100E+02    0.57766E-03
    0.74299E+05    0.86672E+06    0.42687E+04    0.98400E+02    0.38898E-03
    0.53400E+04    0.27007E+06    0.40556E+04    0.99500E+02    0.27991E-03
```

Contiguity matrix R_1 file

Except for the first variant, the first-order autocorrelation is considered in the others. The file for the contiguity matrix R_1 is *R320km.in* . Since the matrix is very big (dimension 120 x 120), only the first two rows is listed below. Note that the name of the matrix (max. 8 characters) should appear on the first line.

In creating this matrix, a rule is used to specify the contiguity or proximity for each O-D pair: if the distance between the origin and the destination of a pair is less than 320 kilometers, then a value of 1 indicating that they are neighbors is assigned to the pair. Otherwise a value of 0 is specified. The Fortran format of the matrix is always (20f3.0) and cannot be changed by the user. It indicates that 20 values are entered on a single line and that the field for each value is 3 characters. Each value entered is right-justified and without a decimal after the point.

```
DIST_OD
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
```

Input file: variant1.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with fixed $\lambda_y = 0$ and one common Box-Cox transformation on the four independent variables POP, REVE, LANGFR and UTILITY with fixed $\lambda_1 = 0$.
- No heteroskedasticity nor autocorrelation is considered.

```
$model nlist=1,nstep=2000,epsi=1.-7,wght=1.,nobs=120,nvaxz=4,nvax=4,norder=0,
nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=0,nexpy=0,nres=0,ncor=0,ncov=0,varname='LOG',nvariant=1$
$yboxcox vly=0.,ntly=0,numlxy=0$
$xboxcox numx=1,2,3,4,numlx=1,1,1,1,vlx=0.,0.,0.,0.,ntlx=0,0,0,0$
$yxzdata yxzfile='Canada1976.in',yxzfmt='(5e15.5)'$
```

Input file: variant2.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with fixed $\lambda_y = 0$ and one common Box-Cox transformation on the four independent variables POP, REVE, LANGFR and UTILITY with fixed $\lambda_1 = 0$.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and fixed $\pi_1 = 1$.
- Display observed, expected values of Y and their differences, and sample values of Y for 120 observations with option nexpy=120.

```
$model nlist=1,nstep=2000,epsi=1.-7,wght=1.,nobs=120,namy='TOTFLOW',nvaxz=4,nvax=4,norder=1,
nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=0,nexpy=120,nres=0,ncor=0,ncov=0,varname='LOG+AU',nvariant=2$
$autocor rho=0.,pi=1.,ntpi=0,matfile='R320km.in'$
$yboxcox vly=0.,ntly=0,numlxy=0$
$xboxcox numx=1,2,3,4,numlx=1,1,1,1,vlx=0.,0.,0.,0.,ntlx=0,0,0,0$
$yxzdata yxzfile='Canada1976.in',yxzfmt='(5e15.5)'$
```

Input file: variant3.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with fixed $\lambda_y = 0$ and one common Box-Cox transformation on the four independent variables POP, REVE, LANGFR and UTILITY with fixed $\lambda_1 = 0$.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.-7,wght=1.,nobs=120,namy='TOTFLOW',nvaxz=4,nvax=4,norder=1,
nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='LOG+AU+PR',nvariant=3$
$autocor rho=0.,pi=0.99,ntpi=1,matfile='R320km.in'$
$yboxcox vly=0.,ntly=0,numlxy=0$
$xboxcox numx=1,2,3,4,numlx=1,1,1,1,vlx=0.,0.,0.,0.,ntlx=0,0,0,0$
$yxzdata yxzfile='Canada1976.in',yxzfmt='(5e15.5)'$
```

Input file: variant4.in

The specification for this variant is as follows:

- One common Box-Cox transformation on the dependent variable Y and the four independent variables POP, REVE, LANGFR and UTILITY with estimated λ_1.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.-7,wght=1.,nobs=120,namy='TOTFLOW',nvaxz=4,nvax=4,norder=1,
nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='BC1+AU+PR',nvariant=4$
$autocor rho=0.,pi=0.99,ntpi=1,matfile='R320km.in'$
$yboxcox vly=0.,ntly=1,numlxy=1$
$xboxcox numx=1,2,3,4,numlx=1,1,1,1,vlx=0.,0.,0.,0.,ntlx=1,0,0,0$
$yxzdata yxzfile='Canada1976.in',yxzfmt='(5e15.5)'$
```

Input file: variant5.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with estimated λ_y and one common Box-Cox transformation on the four independent variables POP, REVE, LANGFR and UTILITY with estimated λ_1.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.-7,wght=1.,nobs=120,namy='TOTFLOW',nvaxz=4,nvax=4,norder=1,
nvaz=0,nlamz=0,nlamy=1,nlamx=1,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='BC2+AU+PR',nvariant=5$
$autocor rho=0.,pi=0.99,ntpi=1,matfile='R320km.in'$
$yboxcox vly=0.,ntly=1,numlxy=0$
$xboxcox numx=1,2,3,4,numlx=1,1,1,1,vlx=0.,0.,0.,0.,ntlx=1,0,0,0$
$yxzdata yxzfile='Canada1976.in',yxzfmt='(5e15.5)'$
```

Input file: variant6.in

The specification for this variant is as follows:

- Box-Cox transformation on the dependent variable Y with estimated λ_y and four distinct Box-Cox transformations on the four independent variables POP, REVE, LANGFR and UTILITY with estimated λ_1, λ_2, λ_3 and λ_4.
- No heteroskedasticity is considered.
- Autocorrelation with estimated ρ_1 and π_1.

```
$model nlist=1,nstep=2000,epsi=1.-7,wght=1.,nobs=120,namy='TOTFLOW',nvaxz=4,nvax=4,norder=1,
nvaz=0,nlamz=0,nlamy=1,nlamx=4,nscale=1,nexpy=0,nres=0,ncor=0,ncov=0,varname='BC5+AU+PR',nvariant=6$
$autocor rho=0.,pi=0.99,ntpi=1,matfile='R320km.in'$
$yboxcox vly=0.,ntly=1,numlxy=0$
$xboxcox numx=1,2,3,4,numlx=1,2,3,4,vlx=0.,0.,0.,0.,ntlx=1,1,1,1$
$yxzdata yxzfile='Canada1976.in',yxzfmt='(5e15.5)'$
```

1.12 Second Example of Output Files

The output files (.txt) which contain the complete results of the variant estimations are stored in the same folder as the input files (.in). For each variant, for example *variant1*, two additional output files *L2Variant1.tab* and *Elasticity1.txt* are also created in the same folder. The former contains the essential results which will be used as input for L-2.1 Tablex program whereas the latter contains the conditional t-values of the coefficients β and the elasticities of $E(Y)$ with respect to the independent variables, which will be used both as inputs for another program called TABQDF which combine total and mode choice results of LEVEL L-2.1 and SHARE S1/S5 or PROBABILITY P2/P6 to obtain modal elasticities and diversion rates in the so-called Quasi Direct Format found in *Liem and Gaudry* (1994).

Output file: variant1.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         120,
 NAMY    = TOTFLOW   ,
 NVAXZ   =           4,
```

```
 NVAX    =           4,
 NORDER  =           0,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           0,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = LOG      ,
 NVARIANT       =           1
 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           0,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           1,           2,           3,           4, 46*0,
 NUMLX   = 4*1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 10*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                                    ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07 -0.4112D+07

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05 -0.5518D+05

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05 -0.1190D+05

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01 -0.8368D+00

CONSTANT    -0.2432D-06 -0.1812D-04 -0.8406D-04 -0.1195D+01  0.1000D+01


INITIAL ESTIMATES

             LAM      PARAMETER         STD-ERROR            STUDENT-T      GRADIENT            DERIV.Y
ELAST.Y
                                                                               DERIV.E(Y)
ELAST.E(Y)


POP                X1   0.1389205D+01   0.5457770D-01       25.4537107  -0.1024327D-05   0.3872510D+00
1.3892049
```

```
                                                                                  0.4970914D+00
1.3892049

REVE              X1   0.8322593D+00   0.6932686D+00        1.2004861  -0.6618732D-06   0.2885581D+02
0.8322593
                                                                                  0.3626500D+02
0.8322593

LANGFR            X1   0.6643968D-01   0.5550501D-01        1.1970033  -0.3276004D-06   0.1338513D+03
0.0664397
                                                                                  0.1916400D+03
0.0664397

UTILITY           X1   0.6335200D+00   0.1496708D-01       42.3275532   0.4788431D-06   0.1902915D+07
0.6335200
                                                                                  0.2843971D+08
0.6335200

CONSTANT        X0  -0.1078358D+02   0.5515421D+01      -1.9551693  -0.7920692D-07  -0.1592291D+07     -
10.7835828
                                                                                 -0.2006817D+07     -
10.7835828

VAR(W)               0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00


LOG-LIKELIHOOD = -0.1318235D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.8850
PEARSON-R2(E) (adjusted for D.F.)   =        0.8810

PSEUDO-R2(L)  (unadjusted)          =        0.9441
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9422

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+06

MEAN of residual W                  = -0.8662091D-10
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00


CPU TIME:      0.02 sec
```

Output file: variant2.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         120,
 NAMY    = TOTFLOW   ,
 NVAXZ   =           4,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           0,
 NEXPY   =         120,
```

```
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = LOG+AU   ,
 NVARIANT        =           2
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
 PI      =   1.00000000000000     ,  0.000000000000000E+000,
 NTPI    =           0,   538976288,
 MATFILE = R320km.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           0,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           1,           2,           3,           4, 46*0,
 NUMLX   = 4*1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 10*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                                   ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
=====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
=====================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07 -0.4112D+07

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05 -0.5518D+05

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05 -0.1190D+05

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01 -0.8368D+00

CONSTANT    -0.2432D-06 -0.1812D-04 -0.8406D-04 -0.1195D+01  0.1000D+01


INITIAL STEP

OBS.NO.  OBS.Y          EXP(Y)       OBS.Y - EXP(Y)   SAMPLE Y
    1  0.3865100D+05  0.1809265D+05  0.2055835D+05  0.1694362D+05
    2  0.7429900D+05  0.3868900D+05  0.3561000D+05  0.3623194D+05
    3  0.5340000D+04  0.5961643D+04 -0.6216431D+03  0.5583031D+04
    4  0.1116600D+05  0.1031070D+05  0.8553005D+03  0.9655887D+04
    5  0.3585000D+04  0.5300975D+04 -0.1715975D+04  0.4964320D+04
    6  0.8164000D+04  0.6446285D+04  0.1717715D+04  0.6036894D+04
    7  0.4554000D+04  0.5405644D+04 -0.8516444D+03  0.5062342D+04
    8  0.7521000D+04  0.6520742D+04  0.1000258D+04  0.6106623D+04
    9  0.6593000D+04  0.5723516D+04  0.8694838D+03  0.5360027D+04
```

10	0.8782000D+04	0.8447647D+04	0.3343529D+03	0.7911153D+04
11	0.2187600D+05	0.2570593D+05	-0.3829932D+04	0.2407340D+05
12	0.1213300D+06	0.1443329D+06	-0.2300294D+05	0.1351666D+06
13	0.8127000D+05	0.1409513D+06	-0.5968126D+05	0.1319997D+06
14	0.1540400D+05	0.1787623D+05	-0.2472228D+04	0.1674094D+05
15	0.2147400D+05	0.2930137D+05	-0.7827370D+04	0.2744050D+05
16	0.1228400D+05	0.1695944D+05	-0.4675443D+04	0.1588238D+05
17	0.1393900D+05	0.1512628D+05	-0.1187278D+04	0.1416564D+05
18	0.1069300D+05	0.1315154D+05	-0.2458538D+04	0.1231631D+05
19	0.1030200D+05	0.1253462D+05	-0.2232615D+04	0.1173857D+05
20	0.5406000D+04	0.7874783D+04	-0.2468783D+04	0.7374671D+04
21	0.4521000D+04	0.8072591D+04	-0.3551591D+04	0.7559916D+04
22	0.2127500D+07	0.2522752D+07	-0.3952520D+06	0.2362537D+07
23	0.2715900D+06	0.2181216D+06	0.5346845D+05	0.2042691D+06
24	0.1517900D+06	0.1296240D+06	0.2216604D+05	0.1213918D+06
25	0.1521500D+07	0.1003390D+07	0.5181096D+06	0.9396670D+06
26	0.7158900D+05	0.5575198D+05	0.1583702D+05	0.5221128D+05
27	0.1650900D+06	0.1434413D+06	0.2164867D+05	0.1343316D+06
28	0.2963400D+06	0.2431084D+06	0.5323162D+05	0.2276690D+06
29	0.2093500D+06	0.1693603D+06	0.3998968D+05	0.1586046D+06
30	0.1498900D+06	0.1251186D+06	0.2477136D+05	0.1171726D+06
31	0.4746000D+05	0.4101675D+05	0.6443245D+04	0.3841186D+05
32	0.1076100D+06	0.9476954D+05	0.1284046D+05	0.8875091D+05
33	0.6450100D+05	0.4996178D+05	0.1453922D+05	0.4678881D+05
34	0.9202700D+05	0.9512321D+05	-0.3096206D+04	0.8908211D+05
35	0.2497100D+05	0.2045990D+05	0.4511097D+04	0.1916053D+05
36	0.3748600D+05	0.4427978D+05	-0.6793782D+04	0.4146766D+05
37	0.8678000D+04	0.1390293D+05	-0.5224933D+04	0.1301998D+05
38	0.6718000D+04	0.1186317D+05	-0.5145169D+04	0.1110976D+05
39	0.3146300D+05	0.2699307D+05	0.4469932D+04	0.2527879D+05
40	0.2328600D+05	0.2762305D+05	-0.4337053D+04	0.2586877D+05
41	0.5486100D+05	0.4278302D+05	0.1207798D+05	0.4006595D+05
42	0.2333700D+06	0.2612121D+06	-0.2784214D+05	0.2446231D+06
43	0.8903600D+05	0.8623637D+05	0.2799626D+04	0.8075967D+05
44	0.8340500D+06	0.7344819D+06	0.9956806D+05	0.6878364D+06
45	0.3720100D+05	0.3567887D+05	0.1522133D+04	0.3341297D+05
46	0.8879500D+05	0.9051852D+05	-0.1723521D+04	0.8476986D+05
47	0.1601100D+06	0.1619257D+06	-0.1815728D+04	0.1516421D+06
48	0.1146300D+06	0.1053250D+06	0.9304971D+04	0.9863604D+05
49	0.8406300D+05	0.7084466D+05	0.1321834D+05	0.6634545D+05
50	0.2168200D+05	0.2168441D+05	-0.2406544D+01	0.2030727D+05
51	0.5095900D+05	0.4953258D+05	0.1426415D+04	0.4638687D+05
52	0.5377700D+05	0.4253678D+05	0.1124022D+05	0.3983535D+05
53	0.6716700D+05	0.6733544D+05	-0.1684433D+03	0.6305910D+05
54	0.1529300D+05	0.1075404D+05	0.4538961D+04	0.1007107D+05
55	0.3626200D+05	0.2400937D+05	0.1225263D+05	0.2248458D+05
56	0.1222400D+05	0.8228885D+04	0.3995115D+04	0.7706285D+04
57	0.9907000D+04	0.6669986D+04	0.3237014D+04	0.6246388D+04
58	0.2310300D+05	0.1349356D+05	0.9609440D+04	0.1263661D+05
59	0.2278700D+05	0.1674567D+05	0.6041331D+04	0.1568218D+05
60	0.3585400D+05	0.2136796D+05	0.1448604D+05	0.2001092D+05
61	0.3893600D+06	0.4442260D+06	-0.5486602D+05	0.4160141D+06
62	0.4266300D+05	0.4209914D+05	0.5638596D+03	0.3942551D+05
63	0.7401600D+05	0.7765977D+05	-0.3643774D+04	0.7272775D+05
64	0.4274600D+05	0.3994097D+05	0.2805029D+04	0.3740440D+05
65	0.3194800D+05	0.2945606D+05	0.2491942D+04	0.2758536D+05
66	0.8775000D+04	0.7526363D+04	0.1248637D+04	0.7048378D+04
67	0.1715600D+05	0.1619422D+05	0.9617814D+03	0.1516576D+05
68	0.1195300D+05	0.1020883D+05	0.1744171D+04	0.9560486D+04
69	0.2736000D+07	0.5655396D+07	-0.2919396D+07	0.5296232D+07
70	0.1289300D+07	0.1871782D+07	-0.5824821D+06	0.1752909D+07
71	0.2139400D+06	0.2178555D+06	-0.3915529D+04	0.2040199D+06
72	0.5161200D+06	0.4546783D+06	0.6144168D+05	0.4258026D+06
73	0.1494000D+06	0.1257645D+06	0.2363554D+05	0.1177774D+06
74	0.2677600D+06	0.2307226D+06	0.3703735D+05	0.2160699D+06
75	0.8032200D+05	0.4301013D+05	0.3731187D+05	0.4027864D+05
76	0.1010400D+06	0.9367755D+05	0.7362455D+04	0.8772827D+05
77	0.2942800D+05	0.3876441D+05	-0.9336413D+04	0.3630256D+05
78	0.2226600D+05	0.3048461D+05	-0.8218609D+04	0.2854859D+05
79	0.7673200D+05	0.5938219D+05	0.1734981D+05	0.5561094D+05
80	0.6434800D+05	0.5817662D+05	0.6171382D+04	0.5448193D+05

81	0.1222000D+06	0.8454028D+05	0.3765972D+05	0.7917129D+05
82	0.6370900D+06	0.1126867D+07	-0.4897768D+06	0.1055302D+07
83	0.7216000D+05	0.7504575D+05	-0.2885752D+04	0.7027974D+05
84	0.1315500D+06	0.1493928D+06	-0.1784282D+05	0.1399052D+06
85	0.2025800D+05	0.1610669D+05	0.4151314D+04	0.1508378D+05
86	0.3354000D+05	0.3156910D+05	0.1970898D+04	0.2956421D+05
87	0.5720000D+04	0.1304642D+05	-0.7326421D+04	0.1221787D+05
88	0.3154000D+04	0.8733989D+04	-0.5579989D+04	0.8179310D+04
89	0.5015000D+04	0.1250024D+05	-0.7485236D+04	0.1170637D+05
90	0.7885000D+06	0.1400739D+07	-0.6122388D+06	0.1311781D+07
91	0.9269500D+05	0.1017410D+06	-0.9046037D+04	0.9527966D+05
92	0.1692700D+06	0.1813698D+06	-0.1209983D+05	0.1698514D+06
93	0.3411500D+05	0.2790925D+05	0.6205754D+04	0.2613678D+05
94	0.5428600D+05	0.5318598D+05	0.1100017D+04	0.4980824D+05
95	0.1421600D+05	0.2206886D+05	-0.7852859D+04	0.2066731D+05
96	0.8656000D+04	0.1408496D+05	-0.5428958D+04	0.1319045D+05
97	0.1496700D+05	0.2000469D+05	-0.5037688D+04	0.1873423D+05
98	0.3195000D+06	0.4518501D+06	-0.1323501D+06	0.4231540D+06
99	0.4823100D+05	0.4525536D+05	0.2975637D+04	0.4238128D+05
100	0.9668800D+05	0.9386816D+05	0.2819845D+04	0.8790677D+05
101	0.1993900D+05	0.1559567D+05	0.4343325D+04	0.1460522D+05
102	0.4677000D+04	0.8046066D+04	-0.3369066D+04	0.7535075D+04
103	0.2577000D+04	0.6491090D+04	-0.3914090D+04	0.6078853D+04
104	0.4218000D+04	0.9085370D+04	-0.4867370D+04	0.8508376D+04
105	0.1490900D+06	0.2114886D+06	-0.6239863D+05	0.1980574D+06
106	0.2599500D+06	0.3475991D+06	-0.8764910D+05	0.3255238D+06
107	0.1497300D+05	0.1243154D+05	0.2541460D+04	0.1164204D+05
108	0.2634000D+05	0.2431636D+05	0.2023639D+04	0.2277208D+05
109	0.9546000D+04	0.1465854D+05	-0.5112541D+04	0.1372761D+05
110	0.6170000D+04	0.8598725D+04	-0.2428725D+04	0.8052637D+04
111	0.9086000D+04	0.1364629D+05	-0.4560293D+04	0.1277964D+05
112	0.2581700D+06	0.5190232D+06	-0.2608532D+06	0.4860611D+06
113	0.7723000D+04	0.7697209D+04	0.2579056D+02	0.7208375D+04
114	0.9061000D+04	0.7886434D+04	0.1174566D+04	0.7385582D+04
115	0.1740200D+05	0.1742530D+05	-0.2330187D+02	0.1631865D+05
116	0.7988000D+04	0.1987733D+05	-0.1188933D+05	0.1861496D+05
117	0.3310000D+04	0.9004813D+04	-0.5694813D+04	0.8432934D+04
118	0.6790000D+04	0.1298105D+05	-0.6191046D+04	0.1215664D+05
119	0.1126800D+06	0.1316980D+06	-0.1901803D+05	0.1233341D+06
120	0.1116000D+05	0.6668290D+04	0.4491710D+04	0.6244799D+04

INITIAL ESTIMATES

	LAM	PARAMETER	STD-ERROR	STUDENT-T	GRADIENT	DERIV.Y DERIV.E(Y)	ELAST.Y ELAST.E(Y)
POP	X1	0.1389205D+01	0.5457770D-01	25.4537107	-0.1024327D-05	0.3872510D+00 0.4970914D+00	1.3892049 1.3892049
REVE	X1	0.8322593D+00	0.6932686D+00	1.2004861	-0.6618732D-06	0.2885581D+02 0.3626500D+02	0.8322593 0.8322593
LANGFR	X1	0.6643968D-01	0.5550501D-01	1.1970033	-0.3276004D-06	0.1338513D+03 0.1916400D+03	0.0664397 0.0664397
UTILITY	X1	0.6335200D+00	0.1496708D-01	42.3275532	0.4788431D-06	0.1902915D+07 0.2843971D+08	0.6335200 0.6335200

```
CONSTANT      X0 -0.1078358D+02  0.5515421D+01     -1.9551693 -0.7920692D-07 -0.1592291D+07    -
10.7835828
                                                                            -0.2006817D+07    -
10.7835828

VAR(W)            0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

RHO1           0.0000000D+00
PI1            0.1000000D+01

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00


LOG-LIKELIHOOD = -0.1318235D+04

NUMBER OF OBSERVATIONS              =            120

PEARSON-R2(E) (unadjusted)          =         0.8850
PEARSON-R2(E) (adjusted for D.F.)   =         0.8810

PSEUDO-R2(L)  (unadjusted)          =         0.9441
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.9422

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+06

MEAN of residual W                  = -0.8662091D-10
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         1      2000   0.1E+01   0.1E-06
  0.00000E+00
  0.10000E+01
  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
   0.1318235024D+04  0.00000D+00

new absolute minimum , parameters for fcn entry     6  follow - - f,x(i)
   0.1294578272D+04  0.96598D+00

step to unallowed region , parameters for fcn entry    11 follow - - x(i)
                     0.15579D+08

step to unallowed region , parameters for fcn entry    12 follow - - x(i)
                     0.77894D+07

step to unallowed region , parameters for fcn entry    13 follow - - x(i)
                     0.38947D+07

step to unallowed region , parameters for fcn entry    14 follow - - x(i)
                     0.19474D+07

new absolute minimum , parameters for fcn entry    15  follow - - f,x(i)
   0.1294578272D+04  0.96598D+00


the run has been terminated since no improvement can be found on the current minimum,
 the parameters of which follow   - - f,x(i)
```

```
   0.1294578272D+04  0.96598D+00


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was   0.1294578272D+04 for entry   19 - - x(i) follows
  0.96598E+00



run statistics follow - -

 nfcn=   21 nst=    4


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP          REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP           0.1000D+01  0.1577D+03  0.1700D+04  0.4335D+07 -0.7709D+07

REVE          0.6340D-02  0.1000D+01  0.1077D+02  0.2748D+05 -0.4887D+05

LANGFR        0.5884D-03  0.9281D-01  0.1000D+01  0.2551D+04 -0.4536D+04

UTILITY       0.2307D-06  0.3639D-04  0.3920D-03  0.1000D+01 -0.1778D+01

CONSTANT     -0.1297D-06 -0.2046D-04 -0.2204D-03 -0.5623D+00  0.1000D+01


FINAL STEP

OBS.NO.  OBS.Y          EXP(Y)       OBS.Y - EXP(Y)   SAMPLE Y
    1  0.3865100D+05  0.2102988D+05  0.1762112D+05  0.2027878D+05
    2  0.7429900D+05  0.4337890D+05  0.3092010D+05  0.5099764D+05
    3  0.5340000D+04  0.6888072D+04 -0.1548072D+04  0.7397103D+04
    4  0.1116600D+05  0.1125550D+05 -0.8950167D+02  0.1248094D+05
    5  0.3585000D+04  0.6216316D+04 -0.2631316D+04  0.6423881D+04
    6  0.8164000D+04  0.7054194D+04  0.1109806D+04  0.7980460D+04
    7  0.4554000D+04  0.6516735D+04 -0.1962735D+04  0.6967064D+04
    8  0.7521000D+04  0.8733075D+04 -0.1212075D+04  0.8421164D+04
    9  0.6593000D+04  0.8471737D+04 -0.1878737D+04  0.8169160D+04
   10  0.8782000D+04  0.1213578D+05 -0.3353782D+04  0.1170234D+05
   11  0.2187600D+05  0.4175095D+05 -0.1987495D+05  0.3039470D+05
   12  0.1213300D+06  0.1222041D+06 -0.8741189D+03  0.1172604D+06
   13  0.8127000D+05  0.1109878D+06 -0.2971779D+05  0.7854723D+05
   14  0.1540400D+05  0.1265770D+05  0.2746298D+04  0.9980082D+04
   15  0.2147400D+05  0.2013581D+05  0.1338187D+04  0.1549820D+05
   16  0.1228400D+05  0.1328432D+05 -0.1000322D+04  0.1005223D+05
   17  0.1393900D+05  0.1020651D+05  0.3732486D+04  0.8123440D+04
   18  0.1069300D+05  0.1233811D+05 -0.1645109D+04  0.8970315D+04
   19  0.1030200D+05  0.1125675D+05 -0.9547538D+03  0.1131208D+05
   20  0.5406000D+04  0.5115730D+04  0.2902696D+03  0.5578028D+04
   21  0.4521000D+04  0.6528830D+04 -0.2007830D+04  0.5465465D+04
   22  0.2127500D+07  0.2760865D+07 -0.6333653D+06  0.2113516D+07
   23  0.2715900D+06  0.2106464D+06  0.6094362D+05  0.1566371D+06
   24  0.1517900D+06  0.1442354D+06  0.7554571D+04  0.1027292D+06
   25  0.1521500D+07  0.1131492D+07  0.3900076D+06  0.8602666D+06
   26  0.7158900D+05  0.5921308D+05  0.1237592D+05  0.4299617D+05
   27  0.1650900D+06  0.1563689D+06  0.8721100D+04  0.1174183D+06
   28  0.2963400D+06  0.2580808D+06  0.3825923D+05  0.1934602D+06
   29  0.2093500D+06  0.1812662D+06  0.2808378D+05  0.1351684D+06
   30  0.1498900D+06  0.1402154D+06  0.9674608D+04  0.1034179D+06
   31  0.4746000D+05  0.4701786D+05  0.4421422D+03  0.3542560D+05
   32  0.1076100D+06  0.1089121D+06 -0.1302112D+04  0.8703195D+05
   33  0.6450100D+05  0.5017356D+05  0.1432744D+05  0.4008386D+05
```

34	0.9202700D+05	0.1086698D+06	-0.1664276D+05	0.9725516D+05
35	0.2497100D+05	0.2327124D+05	0.1699763D+04	0.1908632D+05
36	0.3748600D+05	0.4930228D+05	-0.1181628D+05	0.4418946D+05
37	0.8678000D+04	0.1282208D+05	-0.4144082D+04	0.1485765D+05
38	0.6718000D+04	0.1121411D+05	-0.4496111D+04	0.1254115D+05
39	0.3146300D+05	0.2925458D+05	0.2208418D+04	0.2824636D+05
40	0.2328600D+05	0.3345266D+05	-0.1016666D+05	0.3082916D+05
41	0.5486100D+05	0.4806022D+05	0.6800780D+04	0.4539868D+05
42	0.2333700D+06	0.2454738D+06	-0.1210381D+05	0.2123174D+06
43	0.8903600D+05	0.9382528D+05	-0.4789279D+04	0.8202850D+05
44	0.8340500D+06	0.8610481D+06	-0.2699810D+05	0.7516218D+06
45	0.3720100D+05	0.3781720D+05	-0.6161962D+03	0.3331625D+05
46	0.8879500D+05	0.1005897D+06	-0.1179473D+05	0.8927291D+05
47	0.1601100D+06	0.1747300D+06	-0.1461998D+05	0.1550976D+06
48	0.1146300D+06	0.1134176D+06	0.1212429D+04	0.1002570D+06
49	0.8406300D+05	0.7996101D+05	0.4101989D+04	0.7183763D+05
50	0.2168200D+05	0.2539966D+05	-0.3717663D+04	0.2284747D+05
51	0.5095900D+05	0.5983129D+05	-0.8872292D+04	0.5479397D+05
52	0.5377700D+05	0.4360477D+05	0.1017223D+05	0.3799089D+05
53	0.6716700D+05	0.8757019D+05	-0.2040319D+05	0.7561324D+05
54	0.1529300D+05	0.1560789D+05	-0.3148858D+03	0.1231322D+05
55	0.3626200D+05	0.2650033D+05	0.9761674D+04	0.2889511D+05
56	0.1222400D+05	0.9575937D+04	0.2648063D+04	0.1070912D+05
57	0.9907000D+04	0.7426288D+04	0.2480712D+04	0.8605249D+04
58	0.2310300D+05	0.1900114D+05	0.4101857D+04	0.1751099D+05
59	0.2278700D+05	0.2328504D+05	-0.4980367D+03	0.2248254D+05
60	0.3585400D+05	0.3352706D+05	0.2326940D+04	0.2806643D+05
61	0.3893600D+06	0.4753737D+06	-0.8601368D+05	0.4364522D+06
62	0.4266300D+05	0.4316573D+05	-0.5027317D+03	0.4080883D+05
63	0.7401600D+05	0.8054996D+05	-0.6533957D+04	0.7305501D+05
64	0.4274600D+05	0.3932586D+05	0.3420144D+04	0.3746667D+05
65	0.3194800D+05	0.3066414D+05	0.1283860D+04	0.2977662D+05
66	0.8775000D+04	0.8484780D+04	0.2902196D+03	0.7963539D+04
67	0.1715600D+05	0.1899477D+05	-0.1838769D+04	0.1770047D+05
68	0.1195300D+05	0.1076919D+05	0.1183811D+04	0.1043509D+05
69	0.2736000D+07	0.4616219D+07	-0.1880219D+07	0.4536402D+07
70	0.1289300D+07	0.1688220D+07	-0.3989195D+06	0.1760405D+07
71	0.2139400D+06	0.2242280D+06	-0.1028796D+05	0.2306653D+06
72	0.5161200D+06	0.4766741D+06	0.3944585D+05	0.4987715D+06
73	0.1494000D+06	0.1470010D+06	0.2398983D+04	0.1170826D+06
74	0.2677600D+06	0.2683359D+06	-0.5759077D+03	0.2416162D+06
75	0.8032200D+05	0.5668175D+05	0.2364025D+05	0.5465730D+05
76	0.1010400D+06	0.7119401D+05	0.2984599D+05	0.1226613D+06
77	0.2942800D+05	0.3505295D+05	-0.5624954D+04	0.5539351D+05
78	0.2226600D+05	0.2789134D+05	-0.5625342D+04	0.4313756D+05
79	0.7673200D+05	0.7055249D+05	0.6179506D+04	0.8531860D+05
80	0.6434800D+05	0.4797063D+05	0.1637737D+05	0.8713149D+05
81	0.1222000D+06	0.6505675D+05	0.5714325D+05	0.1239672D+06
82	0.6370900D+06	0.8190817D+06	-0.1819917D+06	0.8295321D+06
83	0.7216000D+05	0.6486337D+05	0.7296630D+04	0.6907066D+05
84	0.1315500D+06	0.1388939D+06	-0.7343950D+04	0.1431363D+06
85	0.2025800D+05	0.1734910D+05	0.2908895D+04	0.1438236D+05
86	0.3354000D+05	0.3423194D+05	-0.6919363D+03	0.3152500D+05
87	0.5720000D+04	0.1015061D+05	-0.4430614D+04	0.1624385D+05
88	0.3154000D+04	0.6588292D+04	-0.3434292D+04	0.1234530D+05
89	0.5015000D+04	0.1073445D+05	-0.5719454D+04	0.1724882D+05
90	0.7885000D+06	0.1043065D+07	-0.2545654D+06	0.1091036D+07
91	0.9269500D+05	0.9012721D+05	0.2567793D+04	0.9358176D+05
92	0.1692700D+06	0.1659731D+06	0.3296857D+04	0.1755607D+06
93	0.3411500D+05	0.2938442D+05	0.4730581D+04	0.2429879D+05
94	0.5428600D+05	0.5688667D+05	-0.2600674D+04	0.5189886D+05
95	0.1421600D+05	0.1579544D+05	-0.1579440D+04	0.2686314D+05
96	0.8656000D+04	0.9614572D+04	-0.9585721D+03	0.1955732D+05
97	0.1496700D+05	0.1542620D+05	-0.4591959D+03	0.2725768D+05
98	0.3195000D+06	0.3353119D+06	-0.1581190D+05	0.3660833D+06
99	0.4823100D+05	0.3954520D+05	0.8685803D+04	0.4241555D+05
100	0.9668800D+05	0.8769219D+05	0.8995809D+04	0.9198910D+05
101	0.1993900D+05	0.1567889D+05	0.4260111D+04	0.1365761D+05
102	0.4677000D+04	0.6171952D+04	-0.1494952D+04	0.1026452D+05
103	0.2577000D+04	0.4816710D+04	-0.2239710D+04	0.9157007D+04
104	0.4218000D+04	0.7577052D+04	-0.3359052D+04	0.1245582D+05

```
  105  0.1490900D+06  0.1643098D+06 -0.1521982D+05  0.1766378D+06
  106  0.2599500D+06  0.2874557D+06 -0.2750571D+05  0.3079677D+06
  107  0.1497300D+05  0.1343888D+05  0.1534120D+04  0.1144424D+05
  108  0.2634000D+05  0.2694800D+05 -0.6080020D+03  0.2501395D+05
  109  0.9546000D+04  0.1067448D+05 -0.1128478D+04  0.1813893D+05
  110  0.6170000D+04  0.5901342D+04  0.2686578D+03  0.1223972D+05
  111  0.9086000D+04  0.1087996D+05 -0.1793964D+04  0.1885352D+05
  112  0.2581700D+06  0.4089880D+06 -0.1508180D+06  0.4071660D+06
  113  0.7723000D+04  0.8689453D+04 -0.9664532D+03  0.8195430D+04
  114  0.9061000D+04  0.9115166D+04 -0.5416584D+02  0.7957117D+04
  115  0.1740200D+05  0.2090816D+05 -0.3506157D+04  0.1948300D+05
  116  0.7988000D+04  0.1442199D+05 -0.6433992D+04  0.2479220D+05
  117  0.3310000D+04  0.6660332D+04 -0.3350332D+04  0.1310565D+05
  118  0.6790000D+04  0.1002370D+05 -0.3233701D+04  0.1830459D+05
  119  0.1126800D+06  0.9865969D+05  0.1402031D+05  0.9569574D+05
  120  0.1116000D+05  0.7573276D+04  0.3586724D+04  0.7302789D+04


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

POP           X1  0.1418847D+01  0.5059234D-01     28.0446973               -0.3303181D-07
0.3955139D+00      1.4188468
                                 0.5059234D-01     28.0446973
0.4505784D+00      1.4188468

REVE          X1  0.1799387D+01  0.7544361D+00      2.3850752               -0.2156735D-07
0.6238772D+02      1.7993868
                                 0.7544361D+00      2.3850752
0.7269231D+02      1.7993868

LANGFR        X1  0.3336479D+00  0.8172845D-01      4.0823961               -0.1096731D-07
0.6721765D+03      0.3336479
                                 0.8172845D-01      4.0823961
0.9141665D+03      0.3336479

UTILITY       X1  0.5708005D+00  0.2277218D-01     25.0656951                0.1764274D-07
0.1714523D+07      0.5708005
                                 0.2277218D-01     25.0656951
0.2623274D+08      0.5708005

CONSTANT      X0 -0.2065001D+02  0.6171206D+01     -3.3461865               -0.2574941D-08 -
0.3049155D+07    -20.6500063
                                 0.6171206D+01     -3.3461865                               -
0.3571227D+07    -20.6500063

VAR(W)            0.7274127D-01  0.1093632D-01      6.6513496                0.0000000D+00
                                 0.1093632D-01      6.6513496

RHO1      (E)     0.7469335D+00  0.6788673D-01     11.0026443                0.4056188D-07
                                 0.6788673D-01     11.0026443
PI1       (F)     0.1000000D+01

LAM(Y)    (F)     0.0000000D+00
LAM(X1)   (F)     0.0000000D+00


LOG-LIKELIHOOD = -0.1294578D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.9433
PEARSON-R2(E) (adjusted for D.F.)   =        0.9408

PSEUDO-R2(L)  (unadjusted)          =        0.9797
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9788
```

```
PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.1729407D+06

MEAN of residual W                  = -0.2064958D-01
VARIANCE of residual W              =  0.7274127D-01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.5492732D+00   Upper Bound =  0.5079740D+00


CPU TIME:      0.30 sec
```

Output file: variant3.txt

```
&MODEL
 NLIST   =          1,
 NSTEP   =       2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =        120,
 NAMY    =  'TOTFLOW',
 NVAXZ   =          4,
 NVAX    =          4,
 NORDER  =          1,
 NVAZ    =          0,
 NLAMZ   =          0,
 NLAMY   =          1,
 NLAMX   =          1,
 NSCALE  =          1,
 NEXPY   =          0,
 NRES    =          0,
 NCOR    =          0,
 NCOV    =          0,
 VARNAME = LOG+AU+PR,
 NVARIANT        =          3
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
 PI      =  0.990000000000000     ,  0.000000000000000E+000,
 NTPI    =          1,          0,
 MATFILE = R320km.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =          0,
 NUMLXY  =          0
 /
 &XBOXCOX
 NUMX    =          1,          2,          3,          4, 46*0,
 NUMLX   = 4*1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 10*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                                   ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01
```

```
MRS of X(l) with respect to X(k) - at the sample means
==============================================================================================================
==============================
       X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
==============================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07  0.1889D+06

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05  0.2536D+04

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05  0.5466D+03

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01  0.3845D-01

CONSTANT     0.5293D-05  0.3944D-03  0.1829D-02  0.2601D+02  0.1000D+01

INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


POP           X1  0.1389205D+01  0.5457770D-01     25.4537107  0.4002112D-09  0.3872510D+00
1.3892049
                                                                              0.4970914D+00
1.3892049

REVE          X1  0.8322593D+00  0.6932686D+00      1.2004861 -0.6521993D-09  0.2885581D+02
0.8322593
                                                                              0.3626500D-01
0.8322593

LANGFR        X1  0.6643968D-01  0.5550501D-01      1.1970033 -0.8357800D-09  0.1338513D+03
0.0664397
                                                                              0.1916400D-02
0.0664397

UTILITY       X1  0.6335200D+00  0.1496708D-01     42.3275532  0.2744231D-08  0.1902915D+07
0.6335200
                                                                              0.2843971D+02
0.6335200

CONSTANT      X0  0.4955077D+00  0.9808520D+00      0.5051808 -0.4526935D-09  0.7316608D+05
0.4955077
                                                                              0.9221363D-01
0.4955077

VAR(W)            0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

RHO1            0.0000000D+00
PI1             0.9900000D+00

LAM(Y)          0.0000000D+00
LAM(X1)         0.0000000D+00


LOG-LIKELIHOOD =  0.3396262D+03

NUMBER OF OBSERVATIONS              =            120

PEARSON-R2(E) (unadjusted)          =         0.8850
PEARSON-R2(E) (adjusted for D.F.)   =         0.8810

PSEUDO-R2(L)  (unadjusted)          =         0.9441
```

```
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.9422

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+00

MEAN of residual W                  = -0.4950736D-12
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         2      2000   0.1E+01   0.1E-06
  0.00000E+00  0.45951E+01
  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.3396262428D+03  0.00000D+00  0.45951D+01

new absolute minimum , parameters for fcn entry     6  follow - - f,x(i)
  -0.3632953214D+03  0.96847D+00  0.45951D+01

new absolute minimum , parameters for fcn entry    10  follow - - f,x(i)
  -0.3633629070D+03  0.96918D+00  0.19391D+01

new absolute minimum , parameters for fcn entry    13  follow - - f,x(i)
  -0.3633635125D+03  0.97006D+00  0.18373D+01

new absolute minimum , parameters for fcn entry    15  follow - - f,x(i)
  -0.3633846711D+03  0.10002D+01  0.17124D+01

new absolute minimum , parameters for fcn entry    18  follow - - f,x(i)
  -0.3633848544D+03  0.10009D+01  0.16626D+01

new absolute minimum , parameters for fcn entry    20  follow - - f,x(i)
  -0.3633848544D+03  0.10008D+01  0.16628D+01

new absolute minimum , parameters for fcn entry    21  follow - - f,x(i)
  -0.3633848544D+03  0.10008D+01  0.16628D+01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3633848544D+03  0.10008D+01  0.16628D+01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3633848544D+03 for entry   21 - - x(i) follows
  0.10008E+01  0.16628E+01



run statistics follow - -

 nfcn=   25 nst=    8


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
```

```
       X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.1490D+03  0.1763D+04  0.4303D+07 -0.7440D+07

REVE         0.6713D-02  0.1000D+01  0.1184D+02  0.2889D+05 -0.4995D+05

LANGFR       0.5671D-03  0.8447D-01  0.1000D+01  0.2440D+04 -0.4219D+04

UTILITY      0.2324D-06  0.3462D-04  0.4098D-03  0.1000D+01 -0.1729D+01

CONSTANT    -0.1344D-06 -0.2002D-04 -0.2370D-03 -0.5784D+00  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

POP           X1  0.1424672D+01  0.5024117D-01     28.3566714                 0.3228170D-08
0.3971378D+00      1.4246723
                                 0.5024117D-01     28.3566714
0.4529296D+00      1.4246723

REVE          X1  0.1706231D+01  0.7686396D+00      2.2198057                 0.2108691D-08
0.5915784D+02      1.7062306
                                 0.7686396D+00      2.2198057
0.6912585D+02      1.7062306

LANGFR        X1  0.3476252D+00  0.8356001D-01      4.1601866                 0.1061520D-08
0.7003356D+03      0.3476252
                                 0.8356001D-01      4.1601866
0.9540261D+03      0.3476252

UTILITY       X1  0.5689552D+00  0.2251792D-01     25.2667725                -0.1708662D-08
0.1708980D+07      0.5689552
                                 0.2251792D-01     25.2667725
0.2615254D+08      0.5689552

CONSTANT      X0 -0.2001134D+02  0.6276615D+01     -3.1882382                 0.2477732D-09 -
0.2954851D+07    -20.0113448
                                 0.6276615D+01     -3.1882382                               -
0.3470618D+07    -20.0113448

VAR(W)            0.6966798D-01  0.1259996D-01      5.5292205                 0.0000000D+00
                                 0.1259996D-01      5.5292205

RHO1       (E)    0.7619495D+00  0.6956915D-01     10.9524055                 0.8854860D-06
                                 0.6956915D-01     10.9524055
PI1        (E)    0.8406104D+00  0.3085371D+00      2.7245029    -0.5165979  0.6019116D-06
                                 0.3085371D+00      2.7245029    -0.5165979

LAM(Y)     (F)    0.0000000D+00
LAM(X1)    (F)    0.0000000D+00


LOG-LIKELIHOOD = -0.1294476D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =         0.9434
PEARSON-R2(E) (adjusted for D.F.)   =         0.9404

PSEUDO-R2(L)  (unadjusted)          =         0.9808
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.9798
```

```
PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1734325D+06

MEAN of residual W                  = -0.2042833D-01
VARIANCE of residual W              =  0.6966798D-01
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.5377643D+00   Upper Bound =  0.4969077D+00


CPU TIME:      0.84 sec
```

Output file: variant4.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         120,
 Namy    =  ‘TOTFLOW’,
 NVAXZ   =           4,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           1,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = BC1+AU+PR,
 NVARIANT        =           4
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
 PI      =  0.990000000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R320km.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           1,
 NUMLXY  =           1
 /
 &XBOXCOX
 NUMX    =           1,           2,           3,           4, 46*0,
 NUMLX   = 4*1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    =           1, 9*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                                   ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
```

```
====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07  0.1889D+06

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05  0.2536D+04

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05  0.5466D+03

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01  0.3845D-01

CONSTANT     0.5293D-05  0.3944D-03  0.1829D-02  0.2601D+02  0.1000D+01

INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


POP           X1  0.1389205D+01  0.5457770D-01     25.4537107  0.4002112D-09  0.3872510D+00
1.3892049
                                                                              0.4970914D+00
1.3892049

REVE          X1  0.8322593D+00  0.6932686D+00      1.2004861 -0.6521993D-09  0.2885581D+02
0.8322593
                                                                              0.3626500D-01
0.8322593

LANGFR        X1  0.6643968D-01  0.5550501D-01      1.1970033 -0.8357800D-09  0.1338513D+03
0.0664397
                                                                              0.1916400D-02
0.0664397

UTILITY       X1  0.6335200D+00  0.1496708D-01     42.3275532  0.2744231D-08  0.1902915D+07
0.6335200
                                                                              0.2843971D+02
0.6335200

CONSTANT      X0  0.4955077D+00  0.9808520D+00      0.5051808 -0.4526935D-09  0.7316608D+05
0.4955077
                                                                              0.9221363D-01
0.4955077

VAR(W)            0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

RHO1           0.0000000D+00
PI1            0.9900000D+00

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00

LAM(Y) is set equal to LAM(X1).


LOG-LIKELIHOOD =  0.3396262D+03

NUMBER OF OBSERVATIONS             =           120

PEARSON-R2(E) (unadjusted)         =        0.8850
PEARSON-R2(E) (adjusted for D.F.)  =        0.8810
```

```
PSEUDO-R2(L)  (unadjusted)          =         0.9441
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.9422

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+00

MEAN of residual W                  = -0.4950736D-12
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00
```

```
initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         3      2000   0.1E+01   0.1E-06
  0.00000E+00  0.45951E+01  0.00000E+00
  0.10000E+01  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.3396262428D+03  0.00000D+00  0.45951D+01  0.00000D+00

new absolute minimum , parameters for fcn entry     4  follow - - f,x(i)
  -0.3411412209D+03  0.15749D-01  0.45951D+01 -0.29047D-01

new absolute minimum , parameters for fcn entry     9  follow - - f,x(i)
  -0.3616140780D+03  0.91950D+00  0.45952D+01 -0.22774D-01

new absolute minimum , parameters for fcn entry    12  follow - - f,x(i)
  -0.3638217837D+03  0.99358D+00  0.45658D+01  0.21078D-01

new absolute minimum , parameters for fcn entry    14  follow - - f,x(i)
  -0.3638515488D+03  0.10269D+01  0.45590D+01  0.21917D-01

new absolute minimum , parameters for fcn entry    16  follow - - f,x(i)
  -0.3638518202D+03  0.10262D+01  0.45229D+01  0.21445D-01

step to unallowed region , parameters for fcn entry    21 follow - - x(i)
                     0.16853D+01 -0.26011D+03 -0.59223D-01

step to unallowed region , parameters for fcn entry    22 follow - - x(i)
                     0.13576D+01 -0.12853D+03 -0.19113D-01

step to unallowed region , parameters for fcn entry    23 follow - - x(i)
                     0.11937D+01 -0.62736D+02  0.94190D-03

step to unallowed region , parameters for fcn entry    24 follow - - x(i)
                     0.11118D+01 -0.29841D+02  0.10969D-01

new absolute minimum , parameters for fcn entry    30  follow - - f,x(i)
  -0.3638826208D+03  0.10317D+01  0.30543D+01  0.20997D-01

new absolute minimum , parameters for fcn entry    37  follow - - f,x(i)
  -0.3639107344D+03  0.10317D+01  0.21380D+01  0.20997D-01

new absolute minimum , parameters for fcn entry    42  follow - - f,x(i)
  -0.3639107450D+03  0.10317D+01  0.21380D+01  0.20909D-01

new absolute minimum , parameters for fcn entry    50  follow - - f,x(i)
  -0.3639144106D+03  0.10429D+01  0.21380D+01  0.20909D-01

new absolute minimum , parameters for fcn entry    58  follow - - f,x(i)
  -0.3639144106D+03  0.10429D+01  0.21380D+01  0.20909D-01

new absolute minimum , parameters for fcn entry    62  follow - - f,x(i)
  -0.3639184186D+03  0.10429D+01  0.19041D+01  0.20909D-01
```

```
new absolute minimum , parameters for fcn entry    67  follow - - f,x(i)
  -0.3639185591D+03  0.10429D+01  0.19041D+01  0.21229D-01

new absolute minimum , parameters for fcn entry    73  follow - - f,x(i)
  -0.3639192867D+03  0.10479D+01  0.19041D+01  0.21229D-01

new absolute minimum , parameters for fcn entry    80  follow - - f,x(i)
  -0.3639198324D+03  0.10499D+01  0.18089D+01  0.21359D-01

new absolute minimum , parameters for fcn entry    84  follow - - f,x(i)
  -0.3639198499D+03  0.10499D+01  0.18089D+01  0.21472D-01

new absolute minimum , parameters for fcn entry    90  follow - - f,x(i)
  -0.3639198549D+03  0.10504D+01  0.18089D+01  0.21472D-01

new absolute minimum , parameters for fcn entry    98  follow - - f,x(i)
  -0.3639198552D+03  0.10503D+01  0.18108D+01  0.21470D-01

new absolute minimum , parameters for fcn entry   104  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18112D+01  0.21496D-01

new absolute minimum , parameters for fcn entry   109  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18112D+01  0.21496D-01

new absolute minimum , parameters for fcn entry   115  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18116D+01  0.21496D-01

new absolute minimum , parameters for fcn entry   122  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   134  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   140  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   146  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   154  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   157  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   165  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   170  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01

new absolute minimum , parameters for fcn entry   185  follow - - f,x(i)
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3639198561D+03  0.10504D+01  0.18117D+01  0.21497D-01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3639198561D+03 for entry  185 - - x(i) follows
  0.10504E+01  0.18117E+01  0.21497E-01
```

```
run statistics follow - -

 nfcn=  187 nst=   29

MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.1624D+03  0.1834D+04  0.4330D+07 -0.7004D+07

REVE         0.6159D-02  0.1000D+01  0.1129D+02  0.2667D+05 -0.4313D+05

LANGFR       0.5454D-03  0.8855D-01  0.1000D+01  0.2362D+04 -0.3820D+04

UTILITY      0.2309D-06  0.3750D-04  0.4234D-03  0.1000D+01 -0.1617D+01

CONSTANT    -0.1428D-06 -0.2318D-04 -0.2618D-03 -0.6183D+00  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

POP           X1  0.1371651D+01  0.6446212D-01     21.2784032                 -0.3124020D-08
0.3930031D+00      1.4098397
                                 0.4575663D-01     29.9771015
0.4242852D+00      1.4406533

REVE          X1  0.1986189D+01  0.8010840D+00      2.4793768                 -0.1915733D-08
0.6381018D+02      1.8404134
                                 0.7930091D+00      2.5046235
0.6830818D+02      1.8906625

LANGFR        X1  0.4212294D+00  0.1083727D+00      3.8868582                 -0.9423495D-09
0.7205804D+03      0.3576741
                                 0.9537639D-01      4.4164952
0.9098762D+03      0.3662355

UTILITY       X1  0.7806878D+00  0.1854142D+00      4.2105078                  0.1359069D-08
0.1701706D+07      0.5665335
                                 0.3003263D-01     25.9946510
0.2462702D+08      0.5453342

CONSTANT      X0 -0.2407558D+02  0.7427566D+01     -3.2413820                 -0.2122847D-09 -
0.2752398D+07    -18.6402568
                                 0.7099785D+01     -3.3910291                                -
0.2961107D+07    -19.1498418

VAR(W)            0.1072360D+00  0.3855558D-01      2.7813363                  0.0000000D+00
                                 0.1966814D-01      5.4522726

RHO1      (E)     0.7819453D+00  0.6206286D-01     12.5992471                  0.3048424D-06
                                 0.6190838D-01     12.6306846
PI1       (E)     0.8595672D+00  0.3123499D+00      2.7519369    -0.4496011  0.2644815D-05
                                 0.3118093D+00      2.7567079    -0.4503805

LAM(Y)    (E)     0.2149724D-01  0.1550819D-01      1.3861867    -63.0958771  0.2379781D-04
LAM(X1)   (E)     0.2149724D-01  0.1550819D-01      1.3861867    -63.0958771  0.2379781D-04

LAM(Y) is set equal to LAM(X1).
```

```
LOG-LIKELIHOOD = -0.1293941D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.9514
PEARSON-R2(E) (adjusted for D.F.)   =        0.9484

PSEUDO-R2(L)  (unadjusted)          =        0.9789
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9776

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1642378D+06

MEAN of residual W                  = -0.2109179D-01
VARIANCE of residual W              =  0.1072360D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.6629313D+00   Upper Bound =  0.6207478D+00


CPU TIME:      4.44 sec
```

Output file: variant5.txt

```
&MODEL
 NLIST   =           1,
 NSTEP   =        2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =         120,
 NAMY    =   'TOTFLOW',
 NVAXZ   =           4,
 NVAX    =           4,
 NORDER  =           1,
 NVAZ    =           0,
 NLAMZ   =           0,
 NLAMY   =           1,
 NLAMX   =           1,
 NSCALE  =           1,
 NEXPY   =           0,
 NRES    =           0,
 NCOR    =           0,
 NCOV    =           0,
 VARNAME = BC2+AU+PR,
 NVARIANT        =          5
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
 PI      =  0.990000000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R320km.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           1,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           1,           2,           3,           4, 46*0,
 NUMLX   = 4*1, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    =           1, 9*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                       ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
```

```
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP          REVE         LANGFR       UTILITY      CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07  0.1889D+06

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05  0.2536D+04

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05  0.5466D+03

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01  0.3845D-01

CONSTANT     0.5293D-05  0.3944D-03  0.1829D-02  0.2601D+02  0.1000D+01


INITIAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T    GRADIENT       DERIV.Y
ELAST.Y
                                                                                DERIV.E(Y)
ELAST.E(Y)


POP           X1  0.1389205D+01  0.5457770D-01     25.4537107  0.4002112D-09  0.3872510D+00
1.3892049
                                                                              0.4970914D+00
1.3892049

REVE          X1  0.8322593D+00  0.6932686D+00      1.2004861 -0.6521993D-09  0.2885581D+02
0.8322593
                                                                              0.3626500D-01
0.8322593

LANGFR        X1  0.6643968D-01  0.5550501D-01      1.1970033 -0.8357800D-09  0.1338513D+03
0.0664397
                                                                              0.1916400D-02
0.0664397

UTILITY       X1  0.6335200D+00  0.1496708D-01     42.3275532  0.2744231D-08  0.1902915D+07
0.6335200
                                                                              0.2843971D+02
0.6335200

CONSTANT      X0  0.4955077D+00  0.9808520D+00      0.5051808 -0.4526935D-09  0.7316608D+05
0.4955077
                                                                              0.9221363D-01
0.4955077

VAR(W)            0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

RHO1           0.0000000D+00
PI1            0.9900000D+00

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00
```

```
LOG-LIKELIHOOD =  0.3396262D+03

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.8850
PEARSON-R2(E) (adjusted for D.F.)   =        0.8810

PSEUDO-R2(L)  (unadjusted)          =        0.9441
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9422

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+00

MEAN of residual W                  = -0.4950736D-12
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         4      2000   0.1E+01   0.1E-06
  0.00000E+00  0.45951E+01  0.00000E+00  0.00000E+00
  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.3396262428D+03  0.00000D+00  0.45951D+01  0.00000D+00  0.00000D+00

new absolute minimum , parameters for fcn entry     5  follow - - f,x(i)
  -0.3574289056D+03  0.10577D-01  0.45951D+01  0.49507D-01 -0.69013D-01

new absolute minimum , parameters for fcn entry     8  follow - - f,x(i)
  -0.3594507225D+03  0.40174D-01  0.45951D+01  0.86369D-01 -0.51569D-01

new absolute minimum , parameters for fcn entry    13  follow - - f,x(i)
  -0.3756384258D+03  0.80630D+00  0.45952D+01  0.47755D-01 -0.73026D-01

new absolute minimum , parameters for fcn entry    17  follow - - f,x(i)
  -0.3768595513D+03  0.85279D+00  0.45629D+01  0.43885D-01 -0.50931D-01

new absolute minimum , parameters for fcn entry    19  follow - - f,x(i)
  -0.3770717099D+03  0.86124D+00  0.45335D+01  0.54802D-01 -0.43083D-01

new absolute minimum , parameters for fcn entry    21  follow - - f,x(i)
  -0.3770827507D+03  0.88333D+00  0.44866D+01  0.54622D-01 -0.43243D-01

new absolute minimum , parameters for fcn entry    30  follow - - f,x(i)
  -0.3776598081D+03  0.91981D+00  0.55148D+00  0.51903D-01 -0.38945D-01

new absolute minimum , parameters for fcn entry    33  follow - - f,x(i)
  -0.3776617176D+03  0.92098D+00  0.45946D+00  0.51871D-01 -0.38878D-01

new absolute minimum , parameters for fcn entry    36  follow - - f,x(i)
  -0.3781389356D+03  0.10032D+01 -0.36572D+00  0.59270D-01 -0.46650D-01

new absolute minimum , parameters for fcn entry    39  follow - - f,x(i)
  -0.3781469117D+03  0.10099D+01 -0.22345D+00  0.58916D-01 -0.46875D-01

new absolute minimum , parameters for fcn entry    41  follow - - f,x(i)
  -0.3781473838D+03  0.10143D+01 -0.24009D+00  0.58769D-01 -0.46809D-01

new absolute minimum , parameters for fcn entry    43  follow - - f,x(i)
  -0.3781473850D+03  0.10143D+01 -0.24097D+00  0.58762D-01 -0.46837D-01
```

```
new absolute minimum , parameters for fcn entry    44  follow - - f,x(i)
  -0.3781473850D+03  0.10143D+01 -0.24095D+00  0.58762D-01 -0.46837D-01

new absolute minimum , parameters for fcn entry    45  follow - - f,x(i)
  -0.3781473850D+03  0.10143D+01 -0.24095D+00  0.58762D-01 -0.46837D-01

new absolute minimum , parameters for fcn entry    47  follow - - f,x(i)
  -0.3781473850D+03  0.10143D+01 -0.24095D+00  0.58762D-01 -0.46837D-01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3781473850D+03  0.10143D+01 -0.24095D+00  0.58762D-01 -0.46837D-01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3781473850D+03 for entry   47 - - x(i) follows
  0.10143E+01 -0.24095E+00  0.58762E-01 -0.46837E-01



run statistics follow - -

 nfcn=   47 nst=   15


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.1013D+03  0.1529D+04  0.4071D+07 -0.1032D+08

REVE         0.9869D-02  0.1000D+01  0.1509D+02  0.4018D+05 -0.1018D+06

LANGFR       0.6542D-03  0.6629D-01  0.1000D+01  0.2663D+04 -0.6750D+04

UTILITY      0.2456D-06  0.2489D-04  0.3754D-03  0.1000D+01 -0.2534D+01

CONSTANT    -0.9692D-07 -0.9821D-05 -0.1482D-03 -0.3946D+00  0.1000D+01


FINAL ESTIMATES

             LAM    PARAMETER      STD-ERROR        STUDENT-T      STUDENT-T    GRADIENT
DERIV.Y        ELAST.Y
                                   (CONDIT.)        (CONDIT.)      (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

POP           X1  0.5141301D+01  0.1451694D+01      3.5415865                -0.1958609D-07
0.3841002D+00      1.3779018
                                 0.1574313D+00     32.6574366
0.3961338D+00      1.5057357

REVE          X1  0.3341642D+01  0.2126171D+01      1.5716717                -0.1406012D-07
0.3892168D+02      1.1225793
                                 0.2027539D+01      1.6481275
0.3659091D+02      1.2110295

LANGFR        X1  0.7172328D+00  0.1774344D+00      4.0422421                -0.7862160D-08
0.5871419D+03      0.2914393
                                 0.1528451D+00      4.6925470
0.6998620D+03      0.3169393
```

```
UTILITY       X1  0.9099308D+00  0.2163702D+00      4.2054350                0.1692249D-07
0.1563835D+07      0.5206335
                                 0.4306024D-01     21.1315780
0.3271641D+08      0.6558488

CONSTANT      X0 -0.5401712D+02  0.1606371D+02     -3.3626802               -0.2022569D-08 -
0.3963081D+07    -26.8394451
                                 0.1342167D+02     -4.0246212                              -
0.3745851D+07    -28.9515672

VAR(W)            0.1791906D+00  0.6357066D-01      2.8187635                0.0000000D+00
                                 0.3167430D-01      5.6572877

RHO1      (E)     0.7675480D+00  0.7667895D-01     10.0098920                0.3714993D-06
                                 0.7163498D-01     10.7147098
PI1       (E)     0.4400516D+00  0.3294861D+00      1.3355693     -1.6994598  0.1051476D-06
                                 0.3216402D+00      1.3681484     -1.7409155

LAM(Y)    (E)     0.5876236D-01  0.1462559D-01      4.0177782    -64.3555467  0.1734071D-05
LAM(X1)   (E)    -0.4683706D-01  0.2383704D-01     -1.9648862    -43.9164106 -0.1729959D-05


LOG-LIKELIHOOD = -0.1279714D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.9710
PEARSON-R2(E) (adjusted for D.F.)   =        0.9689

PSEUDO-R2(L)  (unadjusted)          =        0.9796
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9781

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1498675D+06

MEAN of residual W                  = -0.5989272D-01
VARIANCE of residual W              =  0.1791906D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.8895787D+00   Upper Bound =  0.7697933D+00


CPU TIME:      1.61 sec
```

Output file: variant6.txt

```
&MODEL
 NLIST   =          1,
 NSTEP   =       2000,
 EPSI    =  1.0000000E-07,
 WGHT    =   1.000000    ,
 NOBS    =        120,
 NAMY    =   'TOTFLOW',
 NVAXZ   =          4,
 NVAX    =          4,
 NORDER  =          1,
 NVAZ    =          0,
 NLAMZ   =          0,
 NLAMY   =          1,
 NLAMX   =          4,
 NSCALE  =          1,
 NEXPY   =          0,
 NRES    =          0,
 NCOR    =          0,
 NCOV    =          0,
 VARNAME = BC5+AU+PR,
 NVARIANT        =          6
 /
 &AUTOCOR
 RHO     = 2*0.000000000000000E+000  ,
```

```
 PI      =  0.990000000000000     ,  0.000000000000000E+000,
 NTPI    =           1,           0,
 MATFILE = R320km.in

 /
 &YBOXCOX
 VLY     =  0.000000000000000E+000,
 NTLY    =           1,
 NUMLXY  =           0
 /
 &XBOXCOX
 NUMX    =           1,           2,           3,           4, 46*0,
 NUMLX   =           1,           2,           3,           4, 46*0,
 VLX     = 10*0.000000000000000E+000  ,
 NTLX    = 4*1, 6*0
 /
 &YXZDATA
 YXZFILE = Canada1976.in                                                                   ,
 YXZFMT  = (5e15.5)
 /

VARIABLE       MINIMUM         MEAN          MAXIMUM   NULL OBS.      SCALE
DEPENDENT
TOTFLOW     0.2577000D+04  0.1476588D+06  0.2736000D+07     0     0.1000000D-05
INDEPENDENT
POP         0.8591300D+05  0.5297039D+06  0.2802800D+07     0     0.1000000D-05
REVE        0.3698300D+04  0.4258776D+04  0.4808200D+04     0     0.1000000D-02
LANGFR      0.5600000D+01  0.7329333D+02  0.9990000D+02     0     0.1000000D+00
UTILITY     0.5655600D-04  0.4915871D-01  0.1172800D+01     0     0.1000000D+01
CONSTANT    0.1000000D+01  0.1000000D+01  0.1000000D+01     0     0.1000000D+01


MRS of X(l) with respect to X(k) - at the sample means
====================================================================================================
==============================
        X(k) POP         REVE        LANGFR      UTILITY     CONSTANT
X(l)
====================================================================================================
==============================

POP          0.1000D+01  0.7451D+02  0.3456D+03  0.4914D+07  0.1889D+06

REVE         0.1342D-01  0.1000D+01  0.4639D+01  0.6595D+05  0.2536D+04

LANGFR       0.2893D-02  0.2156D+00  0.1000D+01  0.1422D+05  0.5466D+03

UTILITY      0.2035D-06  0.1516D-04  0.7034D-04  0.1000D+01  0.3845D-01

CONSTANT     0.5293D-05  0.3944D-03  0.1829D-02  0.2601D+02  0.1000D+01


INITIAL ESTIMATES

             LAM      PARAMETER         STD-ERROR            STUDENT-T      GRADIENT           DERIV.Y
ELAST.Y
                                                                               DERIV.E(Y)
ELAST.E(Y)


POP               X1   0.1389205D+01   0.5457770D-01       25.4537107   0.4002112D-09   0.3872510D+00
1.3892049
                                                                             0.4970914D+00
1.3892049

REVE              X2   0.8322593D+00   0.6932686D+00         1.2004861  -0.6521993D-09   0.2885581D+02
0.8322593
                                                                             0.3626500D-01
0.8322593

LANGFR            X3   0.6643968D-01   0.5550501D-01         1.1970033  -0.8357800D-09   0.1338513D+03
0.0664397
```

```
                                                                                         0.1916400D-02
0.0664397

UTILITY          X4   0.6335200D+00   0.1496708D-01       42.3275532   0.2744231D-08   0.1902915D+07
0.6335200
                                                                                         0.2843971D+02
0.6335200

CONSTANT         X0   0.4955077D+00   0.9808520D+00         0.5051808  -0.4526935D-09   0.7316608D+05
0.4955077
                                                                                         0.9221363D-01
0.4955077

VAR(W)             0.1312324D+00  0.1694202D-01      7.7459667  0.0000000D+00

INITIAL VALUES

RHO1           0.0000000D+00
PI1            0.9900000D+00

LAM(Y)         0.0000000D+00
LAM(X1)        0.0000000D+00
LAM(X2)        0.0000000D+00
LAM(X3)        0.0000000D+00
LAM(X4)        0.0000000D+00


LOG-LIKELIHOOD =  0.3396262D+03

NUMBER OF OBSERVATIONS              =            120

PEARSON-R2(E) (unadjusted)          =         0.8850
PEARSON-R2(E) (adjusted for D.F.)   =         0.8810

PSEUDO-R2(L)  (unadjusted)          =         0.9441
PSEUDO-R2(L)  (adjusted for D.F.)   =         0.9422

PROBABILITY of Y to be at the limit =         0.0000
MEAN VALUE of E(Y)                  =  0.1860993D+00

MEAN of residual W                  = -0.4950736D-12
VARIANCE of residual W              =  0.1312324D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7100297D+00   Upper Bound =  0.7100297D+00



initialization cards for control routine follow - -  id card - no. parameters , no. steps , step size
minimization tolerance - starting values of parameters - relative uncertainties of parameters (error
estimates) -
and starting direction  (relative parameter increments for 1st step)
identification line
         7      2000   0.1E+01   0.1E-06
  0.00000E+00  0.45951E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01  0.10000E+01
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

first entry to fcn , parameters of current step follow - - f,x(i)
  -0.3396262428D+03   0.00000D+00   0.45951D+01   0.00000D+00   0.00000D+00   0.00000D+00   0.00000D+00
0.00000D+00

new absolute minimum , parameters for fcn entry     6  follow - - f,x(i)
  -0.3621259716D+03  0.10974D-01  0.45951D+01  0.51368D-01  0.97617D-02  0.82575D-05 -0.61681D-03 -
0.80761D-01

new absolute minimum , parameters for fcn entry    10  follow - - f,x(i)
  -0.3664495955D+03  0.61498D-01  0.45951D+01  0.10106D+00  0.66782D-01  0.22378D-04 -0.50731D-02 -
0.60458D-01

new absolute minimum , parameters for fcn entry    14  follow - - f,x(i)
```

```
  -0.3746705262D+03  0.21468D+00  0.45951D+01  0.83618D-01  0.23957D+00  0.76451D-04 -0.19294D-01 -
0.77137D-01

new absolute minimum , parameters for fcn entry    18  follow - - f,x(i)
  -0.3858983596D+03  0.85768D+00  0.45953D+01  0.53787D-01  0.12043D+00  0.40171D-03 -0.68370D-01 -
0.91861D-01

new absolute minimum , parameters for fcn entry    22  follow - - f,x(i)
  -0.3874669606D+03  0.97621D+00  0.45859D+01  0.46504D-01  0.11897D+00  0.10469D-01  0.48571D+00 -
0.89075D-01

new absolute minimum , parameters for fcn entry    25  follow - - f,x(i)
  -0.3883999604D+03  0.98905D+00  0.45379D+01  0.53030D-01  0.15925D+00  0.68063D-01  0.44704D+00 -
0.71093D-01

new absolute minimum , parameters for fcn entry    27  follow - - f,x(i)
  -0.3885541365D+03  0.10090D+01  0.44725D+01  0.59508D-01  0.15088D+00  0.15666D+00  0.46152D+00 -
0.65107D-01

new absolute minimum , parameters for fcn entry    30  follow - - f,x(i)
  -0.3885963613D+03  0.10414D+01  0.42441D+01  0.57507D-01  0.14628D+00  0.45933D+00  0.41044D+00 -
0.65535D-01

step to unallowed region , parameters for fcn entry    34 follow - - x(i)
                     0.80868D+01 -0.18146D+03  0.28890D+00  0.87223D+00  0.22461D+03  0.17164D+01 -
0.36103D+00

step to unallowed region , parameters for fcn entry    35 follow - - x(i)
                     0.46262D+01 -0.90247D+02  0.17524D+00  0.51566D+00  0.11451D+03  0.10749D+01 -
0.21589D+00

step to unallowed region , parameters for fcn entry    36 follow - - x(i)
                     0.28960D+01 -0.44639D+02  0.11842D+00  0.33737D+00  0.59461D+02  0.75421D+00 -
0.14332D+00

step to unallowed region , parameters for fcn entry    37 follow - - x(i)
                     0.20308D+01 -0.21836D+02  0.90002D-01  0.24823D+00  0.31937D+02  0.59384D+00 -
0.10703D+00

new absolute minimum , parameters for fcn entry    43  follow - - f,x(i)
  -0.3905713535D+03  0.11438D+01  0.96830D+00  0.61589D-01  0.15909D+00  0.44132D+01  0.43348D+00 -
0.70748D-01

new absolute minimum , parameters for fcn entry    47  follow - - f,x(i)
  -0.3916544619D+03  0.11438D+01 -0.26180D+01  0.61589D-01  0.15909D+00  0.44132D+01  0.43348D+00 -
0.70748D-01

new absolute minimum , parameters for fcn entry    55  follow - - f,x(i)
  -0.3917460494D+03  0.11438D+01 -0.26180D+01  0.66990D-01  0.15909D+00  0.44132D+01  0.43348D+00 -
0.70748D-01

new absolute minimum , parameters for fcn entry    60  follow - - f,x(i)
  -0.3917568353D+03  0.11438D+01 -0.26180D+01  0.66990D-01  0.16463D+00  0.44132D+01  0.43348D+00 -
0.70748D-01

step to unallowed region , parameters for fcn entry    66 follow - - x(i)
                     0.11438D+01 -0.26180D+01  0.66990D-01  0.16463D+00  0.43803D+02  0.43348D+00 -
0.70748D-01

step to unallowed region , parameters for fcn entry    67 follow - - x(i)
                     0.11438D+01 -0.26180D+01  0.66990D-01  0.16463D+00  0.39421D+02  0.43348D+00 -
0.70748D-01

new absolute minimum , parameters for fcn entry    65  follow - - f,x(i)
  -0.3926058959D+03  0.11438D+01 -0.26180D+01  0.66990D-01  0.16463D+00  0.17507D+02  0.43348D+00 -
0.70748D-01

new absolute minimum , parameters for fcn entry    77  follow - - f,x(i)
  -0.3939715332D+03  0.11438D+01 -0.26180D+01  0.66990D-01  0.16463D+00  0.17507D+02 -0.56201D-01 -
0.70748D-01
```

```
new absolute minimum , parameters for fcn entry    84  follow - - f,x(i)
  -0.3939857700D+03   0.11438D+01  -0.26180D+01   0.66990D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry    90  follow - - f,x(i)
  -0.3942591923D+03   0.12692D+01  -0.26180D+01   0.66990D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry    98  follow - - f,x(i)
  -0.3942592011D+03   0.12699D+01  -0.26180D+01   0.66990D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

step to unallowed region , parameters for fcn entry   103 follow - - x(i)
                      0.12699D+01  -0.49096D+02   0.66990D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

step to unallowed region , parameters for fcn entry   104 follow - - x(i)
                      0.12699D+01  -0.43867D+02   0.66990D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry   102  follow - - f,x(i)
  -0.3944848250D+03   0.12699D+01  -0.17723D+02   0.66990D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry   111  follow - - f,x(i)
  -0.3944850033D+03   0.12699D+01  -0.17723D+02   0.66760D-01   0.16463D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry   116  follow - - f,x(i)
  -0.3944856580D+03   0.12699D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.17507D+02  -0.56201D-01  -
0.68365D-01

step to unallowed region , parameters for fcn entry   118 follow - - x(i)
                      0.12699D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.21715D+02  -0.56201D-01  -
0.68365D-01

step to unallowed region , parameters for fcn entry   121 follow - - x(i)
                      0.12699D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.20200D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry   122  follow - - f,x(i)
  -0.3946270694D+03   0.12699D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.56201D-01  -
0.68365D-01

new absolute minimum , parameters for fcn entry   126  follow - - f,x(i)
  -0.3947120456D+03   0.12699D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.68365D-01

new absolute minimum , parameters for fcn entry   130  follow - - f,x(i)
  -0.3947965239D+03   0.12699D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.73999D-01

new absolute minimum , parameters for fcn entry   133  follow - - f,x(i)
  -0.3948011549D+03   0.12861D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.73999D-01

new absolute minimum , parameters for fcn entry   138  follow - - f,x(i)
  -0.3948011550D+03   0.12862D+01  -0.17723D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.73999D-01

step to unallowed region , parameters for fcn entry   141 follow - - x(i)
                      0.12862D+01  -0.37802D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.73999D-01

step to unallowed region , parameters for fcn entry   142 follow - - x(i)
                      0.12862D+01  -0.35292D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.73999D-01

new absolute minimum , parameters for fcn entry   140  follow - - f,x(i)
  -0.3948011625D+03   0.12862D+01  -0.22743D+02   0.66760D-01   0.16331D+00   0.19863D+02  -0.18388D+00  -
0.73999D-01
```

```
new absolute minimum , parameters for fcn entry   149  follow - - f,x(i)
  -0.3948358176D+03  0.12862D+01 -0.22743D+02  0.63563D-01  0.16331D+00  0.19863D+02 -0.18388D+00 -
0.73999D-01

new absolute minimum , parameters for fcn entry   155  follow - - f,x(i)
  -0.3948414950D+03  0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.19863D+02 -0.18388D+00 -
0.73999D-01

step to unallowed region , parameters for fcn entry   157 follow - - x(i)
                     0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.20187D+02 -0.18388D+00 -
0.73999D-01

step to unallowed region , parameters for fcn entry   160 follow - - x(i)
                     0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.20070D+02 -0.18388D+00 -
0.73999D-01

step to unallowed region , parameters for fcn entry   161 follow - - x(i)
                     0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.20044D+02 -0.18388D+00 -
0.73999D-01

new absolute minimum , parameters for fcn entry   159  follow - - f,x(i)
  -0.3948442253D+03  0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.18388D+00 -
0.73999D-01

new absolute minimum , parameters for fcn entry   166  follow - - f,x(i)
  -0.3948467194D+03  0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.73999D-01

new absolute minimum , parameters for fcn entry   168  follow - - f,x(i)
  -0.3948656851D+03  0.12862D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

new absolute minimum , parameters for fcn entry   177  follow - - f,x(i)
  -0.3948661138D+03  0.12911D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

new absolute minimum , parameters for fcn entry   186  follow - - f,x(i)
  -0.3948661138D+03  0.12911D+01 -0.22743D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

step to unallowed region , parameters for fcn entry   188 follow - - x(i)
                     0.12911D+01 -0.25152D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

step to unallowed region , parameters for fcn entry   190 follow - - x(i)
                     0.12911D+01 -0.23129D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

new absolute minimum , parameters for fcn entry   194  follow - - f,x(i)
  -0.3948661204D+03  0.12911D+01 -0.22753D+02  0.63563D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

new absolute minimum , parameters for fcn entry   200  follow - - f,x(i)
  -0.3948706517D+03  0.12911D+01 -0.22753D+02  0.62406D-01  0.15943D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

new absolute minimum , parameters for fcn entry   204  follow - - f,x(i)
  -0.3948711047D+03  0.12911D+01 -0.22753D+02  0.62406D-01  0.15834D+00  0.19915D+02 -0.16283D+00 -
0.76661D-01

step to unallowed region , parameters for fcn entry   207 follow - - x(i)
                     0.12911D+01 -0.22753D+02  0.62406D-01  0.15834D+00  0.20014D+02 -0.16283D+00 -
0.76661D-01

step to unallowed region , parameters for fcn entry   208 follow - - x(i)
                     0.12911D+01 -0.22753D+02  0.62406D-01  0.15834D+00  0.20002D+02 -0.16283D+00 -
0.76661D-01

new absolute minimum , parameters for fcn entry   206  follow - - f,x(i)
```

```
  -0.3948724064D+03   0.12911D+01  -0.22753D+02   0.62406D-01   0.15834D+00   0.19940D+02  -0.16283D+00  -
0.76661D-01

new absolute minimum , parameters for fcn entry   212  follow - - f,x(i)
  -0.3948728470D+03   0.12911D+01  -0.22753D+02   0.62406D-01   0.15834D+00   0.19940D+02  -0.15393D+00  -
0.76661D-01

new absolute minimum , parameters for fcn entry   216  follow - - f,x(i)
  -0.3948752346D+03   0.12911D+01  -0.22753D+02   0.62406D-01   0.15834D+00   0.19940D+02  -0.15393D+00  -
0.77606D-01

new absolute minimum , parameters for fcn entry   222  follow - - f,x(i)
  -0.3948752805D+03   0.12927D+01  -0.22753D+02   0.62406D-01   0.15834D+00   0.19940D+02  -0.15393D+00  -
0.77606D-01

new absolute minimum , parameters for fcn entry   229  follow - - f,x(i)
  -0.3948767767D+03   0.12943D+01  -0.22763D+02   0.61248D-01   0.15724D+00   0.19965D+02  -0.14503D+00  -
0.78551D-01

new absolute minimum , parameters for fcn entry   232  follow - - f,x(i)
  -0.3948767824D+03   0.12943D+01  -0.22782D+02   0.61248D-01   0.15724D+00   0.19965D+02  -0.14503D+00  -
0.78551D-01

new absolute minimum , parameters for fcn entry   237  follow - - f,x(i)
  -0.3948772131D+03   0.12943D+01  -0.22782D+02   0.61605D-01   0.15724D+00   0.19965D+02  -0.14503D+00  -
0.78551D-01

new absolute minimum , parameters for fcn entry   242  follow - - f,x(i)
  -0.3948772568D+03   0.12943D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.19965D+02  -0.14503D+00  -
0.78551D-01

step to unallowed region , parameters for fcn entry   245 follow - - x(i)
                     0.12943D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.20012D+02  -0.14503D+00  -
0.78551D-01

step to unallowed region , parameters for fcn entry   246 follow - - x(i)
                     0.12943D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.20006D+02  -0.14503D+00  -
0.78551D-01

new absolute minimum , parameters for fcn entry   244  follow - - f,x(i)
  -0.3948778755D+03   0.12943D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.19977D+02  -0.14503D+00  -
0.78551D-01

new absolute minimum , parameters for fcn entry   251  follow - - f,x(i)
  -0.3948779067D+03   0.12943D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.19977D+02  -0.14741D+00  -
0.78551D-01

new absolute minimum , parameters for fcn entry   256  follow - - f,x(i)
  -0.3948781423D+03   0.12943D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.19977D+02  -0.14741D+00  -
0.78254D-01

new absolute minimum , parameters for fcn entry   263  follow - - f,x(i)
  -0.3948781464D+03   0.12938D+01  -0.22782D+02   0.61605D-01   0.15758D+00   0.19977D+02  -0.14741D+00  -
0.78254D-01

new absolute minimum , parameters for fcn entry   269  follow - - f,x(i)
  -0.3948787746D+03   0.12933D+01  -0.22801D+02   0.61962D-01   0.15792D+00   0.19989D+02  -0.14979D+00  -
0.77957D-01

new absolute minimum , parameters for fcn entry   272  follow - - f,x(i)
  -0.3948787876D+03   0.12933D+01  -0.22765D+02   0.61962D-01   0.15792D+00   0.19989D+02  -0.14979D+00  -
0.77957D-01

new absolute minimum , parameters for fcn entry   277  follow - - f,x(i)
  -0.3948788232D+03   0.12933D+01  -0.22765D+02   0.61859D-01   0.15792D+00   0.19989D+02  -0.14979D+00  -
0.77957D-01

new absolute minimum , parameters for fcn entry   282  follow - - f,x(i)
  -0.3948788269D+03   0.12933D+01  -0.22765D+02   0.61859D-01   0.15782D+00   0.19989D+02  -0.14979D+00  -
0.77957D-01
```

```
step to unallowed region , parameters for fcn entry   285 follow - - x(i)
                     0.12933D+01 -0.22765D+02  0.61859D-01  0.15782D+00  0.20011D+02 -0.14979D+00 -
0.77957D-01

step to unallowed region , parameters for fcn entry   286 follow - - x(i)
                     0.12933D+01 -0.22765D+02  0.61859D-01  0.15782D+00  0.20009D+02 -0.14979D+00 -
0.77957D-01

new absolute minimum , parameters for fcn entry   284  follow - - f,x(i)
  -0.3948791229D+03  0.12933D+01 -0.22765D+02  0.61859D-01  0.15782D+00  0.19994D+02 -0.14979D+00 -
0.77957D-01

new absolute minimum , parameters for fcn entry   291  follow - - f,x(i)
  -0.3948791234D+03  0.12933D+01 -0.22765D+02  0.61859D-01  0.15782D+00  0.19994D+02 -0.14950D+00 -
0.77957D-01

new absolute minimum , parameters for fcn entry   296  follow - - f,x(i)
  -0.3948791459D+03  0.12933D+01 -0.22765D+02  0.61859D-01  0.15782D+00  0.19994D+02 -0.14950D+00 -
0.78048D-01

new absolute minimum , parameters for fcn entry   300  follow - - f,x(i)
  -0.3948791467D+03  0.12935D+01 -0.22765D+02  0.61859D-01  0.15782D+00  0.19994D+02 -0.14950D+00 -
0.78048D-01

new absolute minimum , parameters for fcn entry   308  follow - - f,x(i)
  -0.3948794401D+03  0.12937D+01 -0.22729D+02  0.61757D-01  0.15773D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

new absolute minimum , parameters for fcn entry   309  follow - - f,x(i)
  -0.3948794473D+03  0.12937D+01 -0.22750D+02  0.61757D-01  0.15773D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

new absolute minimum , parameters for fcn entry   316  follow - - f,x(i)
  -0.3948794486D+03  0.12937D+01 -0.22750D+02  0.61777D-01  0.15773D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

new absolute minimum , parameters for fcn entry   321  follow - - f,x(i)
  -0.3948794487D+03  0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

step to unallowed region , parameters for fcn entry   323 follow - - x(i)
                     0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20003D+02 -0.14921D+00 -
0.78140D-01

step to unallowed region , parameters for fcn entry   325 follow - - x(i)
                     0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

step to unallowed region , parameters for fcn entry   327 follow - - x(i)
                     0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

step to unallowed region , parameters for fcn entry   330 follow - - x(i)
                     0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

step to unallowed region , parameters for fcn entry   331 follow - - x(i)
                     0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

new absolute minimum , parameters for fcn entry   329  follow - - f,x(i)
  -0.3948794493D+03  0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14921D+00 -
0.78140D-01

new absolute minimum , parameters for fcn entry   335  follow - - f,x(i)
  -0.3948794498D+03  0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14891D+00 -
0.78140D-01

new absolute minimum , parameters for fcn entry   341  follow - - f,x(i)
  -0.3948794512D+03  0.12937D+01 -0.22750D+02  0.61777D-01  0.15774D+00  0.20000D+02 -0.14891D+00 -
0.78117D-01
```

```
new absolute minimum , parameters for fcn entry   347  follow - - f,x(i)
  -0.3948794513D+03   0.12937D+01 -0.22750D+02   0.61777D-01   0.15774D+00   0.20000D+02 -0.14891D+00 -
0.78117D-01

new absolute minimum , parameters for fcn entry   353  follow - - f,x(i)
  -0.3948794575D+03   0.12936D+01 -0.22756D+02   0.61783D-01   0.15775D+00   0.20000D+02 -0.14882D+00 -
0.78110D-01

new absolute minimum , parameters for fcn entry   357  follow - - f,x(i)
  -0.3948794578D+03   0.12936D+01 -0.22756D+02   0.61792D-01   0.15775D+00   0.20000D+02 -0.14882D+00 -
0.78110D-01

new absolute minimum , parameters for fcn entry   362  follow - - f,x(i)
  -0.3948794579D+03   0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14882D+00 -
0.78110D-01

step to unallowed region , parameters for fcn entry   365 follow - - x(i)
                     0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14882D+00 -
0.78110D-01

step to unallowed region , parameters for fcn entry   366 follow - - x(i)
                     0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14882D+00 -
0.78110D-01

new absolute minimum , parameters for fcn entry   364  follow - - f,x(i)
  -0.3948794582D+03   0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14882D+00 -
0.78110D-01

new absolute minimum , parameters for fcn entry   369  follow - - f,x(i)
  -0.3948794583D+03   0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14900D+00 -
0.78110D-01

new absolute minimum , parameters for fcn entry   376  follow - - f,x(i)
  -0.3948794584D+03   0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14900D+00 -
0.78103D-01

new absolute minimum , parameters for fcn entry   384  follow - - f,x(i)
  -0.3948794584D+03   0.12936D+01 -0.22756D+02   0.61792D-01   0.15776D+00   0.20000D+02 -0.14900D+00 -
0.78103D-01

step to unallowed region , parameters for fcn entry   385 follow - - x(i)
                     0.12935D+01 -0.22777D+02   0.61812D-01   0.15778D+00   0.20000D+02 -0.14870D+00 -
0.78080D-01

new absolute minimum , parameters for fcn entry   389  follow - - f,x(i)
  -0.3948794608D+03   0.12936D+01 -0.22759D+02   0.61795D-01   0.15776D+00   0.20000D+02 -0.14895D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   405  follow - - f,x(i)
  -0.3948794640D+03   0.12936D+01 -0.22763D+02   0.61808D-01   0.15778D+00   0.20000D+02 -0.14908D+00 -
0.78090D-01

new absolute minimum , parameters for fcn entry   413  follow - - f,x(i)
  -0.3948794642D+03   0.12936D+01 -0.22763D+02   0.61800D-01   0.15778D+00   0.20000D+02 -0.14908D+00 -
0.78090D-01

new absolute minimum , parameters for fcn entry   416  follow - - f,x(i)
  -0.3948794642D+03   0.12936D+01 -0.22763D+02   0.61800D-01   0.15777D+00   0.20000D+02 -0.14908D+00 -
0.78090D-01

step to unallowed region , parameters for fcn entry   418 follow - - x(i)
                     0.12936D+01 -0.22763D+02   0.61800D-01   0.15777D+00   0.20000D+02 -0.14908D+00 -
0.78090D-01

step to unallowed region , parameters for fcn entry   422 follow - - x(i)
                     0.12936D+01 -0.22763D+02   0.61800D-01   0.15777D+00   0.20000D+02 -0.14908D+00 -
0.78090D-01

step to unallowed region , parameters for fcn entry   423 follow - - x(i)
```

```
                     0.12936D+01 -0.22763D+02  0.61800D-01  0.15777D+00  0.20000D+02 -0.14908D+00 -
0.78090D-01

new absolute minimum , parameters for fcn entry   421  follow - - f,x(i)
  -0.3948794643D+03  0.12936D+01 -0.22763D+02  0.61800D-01  0.15777D+00  0.20000D+02 -0.14908D+00 -
0.78090D-01

new absolute minimum , parameters for fcn entry   430  follow - - f,x(i)
  -0.3948794643D+03  0.12936D+01 -0.22763D+02  0.61800D-01  0.15777D+00  0.20000D+02 -0.14907D+00 -
0.78090D-01

new absolute minimum , parameters for fcn entry   433  follow - - f,x(i)
  -0.3948794644D+03  0.12936D+01 -0.22763D+02  0.61800D-01  0.15777D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

new absolute minimum , parameters for fcn entry   438  follow - - f,x(i)
  -0.3948794644D+03  0.12936D+01 -0.22763D+02  0.61800D-01  0.15777D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

step to unallowed region , parameters for fcn entry   440 follow - - x(i)
                     0.12936D+01 -0.22764D+02  0.61802D-01  0.15777D+00  0.20000D+02 -0.14905D+00 -
0.78095D-01

new absolute minimum , parameters for fcn entry   459  follow - - f,x(i)
  -0.3948794644D+03  0.12936D+01 -0.22763D+02  0.61800D-01  0.15777D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

new absolute minimum , parameters for fcn entry   465  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15777D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

new absolute minimum , parameters for fcn entry   472  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

step to unallowed region , parameters for fcn entry   474 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

step to unallowed region , parameters for fcn entry   478 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

step to unallowed region , parameters for fcn entry   479 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

new absolute minimum , parameters for fcn entry   477  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14907D+00 -
0.78097D-01

new absolute minimum , parameters for fcn entry   485  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78097D-01

new absolute minimum , parameters for fcn entry   489  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   493  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   498  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   505  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61797D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01
```

```
new absolute minimum , parameters for fcn entry   511  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   515  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

step to unallowed region , parameters for fcn entry   518 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

step to unallowed region , parameters for fcn entry   519 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   517  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14905D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   523  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78100D-01

new absolute minimum , parameters for fcn entry   527  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   531  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   533  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   539  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   545  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   552  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

step to unallowed region , parameters for fcn entry   554 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

step to unallowed region , parameters for fcn entry   557 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

step to unallowed region , parameters for fcn entry   558 follow - - x(i)
                     0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   556  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14904D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   562  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14903D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   568  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14903D+00 -
0.78101D-01
```

```
new absolute minimum , parameters for fcn entry   570  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14903D+00 -
0.78101D-01

new absolute minimum , parameters for fcn entry   573  follow - - f,x(i)
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14903D+00 -
0.78101D-01


the function (fcn) has not improved by    0.10000D-06 after 3 steps which include a minimum - - -


therefore the run has been terminated
  -0.3948794645D+03  0.12936D+01 -0.22763D+02  0.61796D-01  0.15776D+00  0.20000D+02 -0.14903D+00 -
0.78101D-01


all numbers of last entry refer to final pt. of run


the lowest value of the function (fcn) was  -0.3948794645D+03 for entry  573 - - x(i) follows
  0.12936E+01 -0.22763E+02  0.61796E-01  0.15776E+00  0.20000E+02 -0.14903E+00 -0.78101E-01



run statistics follow - -

 nfcn=  575 nst=  104
```

MRS of X(l) with respect to X(k) - at the sample means

X(k) / X(l)	POP	REVE	LANGFR	UTILITY	CONSTANT
POP	0.1000D+01	0.5343D+02	0.9507D+03	0.3555D+07	0.5750D+06
REVE	0.1872D-01	0.1000D+01	0.1779D+02	0.6654D+05	0.1076D+05
LANGFR	0.1052D-02	0.5620D-01	0.1000D+01	0.3739D+04	0.6048D+03
UTILITY	0.2813D-06	0.1503D-04	0.2674D-03	0.1000D+01	0.1617D+00
CONSTANT	0.1739D-05	0.9292D-04	0.1653D-02	0.6183D+01	0.1000D+01

FINAL ESTIMATES

```
             LAM      PARAMETER         STD-ERROR           STUDENT-T         STUDENT-T      GRADIENT
DERIV.Y         ELAST.Y
                                   (CONDIT.)                         (CONDIT.)                 (CONDIT.)
DERIV.E(Y)      ELAST.E(Y)
                                                    (PAR = 0)      (PAR = 1)

POP               X1   0.3844694D+00   0.3517546D+00        1.0930047                      -0.6247733D-10
0.4108518D+00      1.4738693
                                 0.1025131D-01                                                 37.5044123
0.3878941D+00      1.5402461

REVE              X2   0.3429527D-72   0.4665608D-70        0.0073507                       0.1974488D+60
0.2195166D+02      0.6331298
                                 0.1536530D-72                                                  2.2319941
0.3198664D+02      1.1378275

LANGFR            X3   0.7672183D+00   0.8228447D+00        0.9323975                      -0.1311773D-11
0.3905881D+03      0.1938760
```

```
                                0.1147091D+00                                             6.6883801
0.5236958D+03      0.2164130

UTILITY         X4   0.8018848D+00   0.2973747D+00        2.6965467                       0.1545454D-10
0.1460565D+07      0.4862527
                                0.3290807D-01                                            24.3674229
0.3883146D+08      0.6869799

CONSTANT        X0   0.3338211D+01   0.4918092D+01        0.6787614                      -0.1589866D-11
0.2362306D+06      1.5998408
                                0.8761889D+00                                             3.8099218
0.2247487D+06      1.7331927

VAR(W)             0.1347394D+00  0.7513836D-01      1.7932166                0.0000000D+00
                                  0.1977638D-01      6.8131465

RHO1      (E)      0.8600723D+00  0.1329730D+00      6.4680218               -0.1557060D-05
                                  0.1262654D+00      6.8116242
PI1       (E)      0.1300980D-09  0.2781831D-04      0.0000047 -35947.5469534 -0.8229619D+05
                                  0.2640790D-04      0.0000049 -37867.4560232

LAM(Y)    (E)      0.6179552D-01  0.2576128D-01      2.3987753    -36.4191743 -0.9091186D-03
LAM(X1)   (E)      0.1577619D+00  0.6149874D-01      2.5652863    -13.6952083 -0.6548078D-04
LAM(X2)   (E)      0.2000000D+02  0.1614478D+02      1.2387907      1.1768511  0.5170128D-01
LAM(X3)   (E)     -0.1490339D+00  0.3466592D+00     -0.4299146     -3.3145918  0.1723611D-04
LAM(X4)   (E)     -0.7810083D-01  0.2410603D-01     -3.2398883    -44.7232930 -0.1042635D-03


LOG-LIKELIHOOD = -0.1262982D+04

NUMBER OF OBSERVATIONS              =           120

PEARSON-R2(E) (unadjusted)          =        0.9962
PEARSON-R2(E) (adjusted for D.F.)   =        0.9959

PSEUDO-R2(L)  (unadjusted)          =        0.9888
PSEUDO-R2(L)  (adjusted for D.F.)   =        0.9876

PROBABILITY of Y to be at the limit =        0.0000
MEAN VALUE of E(Y)                  =  0.1513731D+06

MEAN of residual W                  = -0.8002553D-01
VARIANCE of residual W              =  0.1347394D+00
95% CONFIDENCE INTERVAL for residual W
Lower Bound = -0.7994800D+00   Upper Bound =  0.6394289D+00


CPU TIME:      13.91 sec
```

2. LEVEL L-2.1 TABLEX

- L-2.1 TABLEX is a program which generates a report of a set of selected variants estimated by the L-2.1 program in tabular form that can be imported into a Word document. Nine statistics such as the regression coefficients, their conditional t-statistics and the derivatives and elasticities of the dependent variable with respect to the independent variables, among others, are available.

2.1 Input File

-

The input file for L-2.1 TABLEX contains three sections:

SECTION I : Tablex options	
First Card (List Options) **_$card nbloc=…,nmatr=…,neq=…,ndt=…,nvs=…,nlist=…,nstat=…$** (Symbol _ indicates a blank. A value should be specified for each parameter after the equality sign)	
nbloc	Number of blocks of independent variables X in the list of names and definitions of the blocks and the variables given in the file FNAME declared in Section II. The number of blocks is unlimited with a total of 350 independent variables.
nmatr	Number of contiguity matrices R used in the NEQ equations (max. 10).
neq	Number of equations (or variants) estimated by L-2.1 program, to be reported in Tablex (max. 10).
ndt	Number of variables Z (or Delta coefficients) used for heteroskedasticity in the NEQ equations (max. 5).
nvs	Language version of the tablex (nvs=0 for French, nvs=1 for English).
nlist	Output option for TABLEX (nlist=0 for a normal run where only the tablex is produced, nlist=1 when only the list of blocks with the definitions of the independent variables X is produced).
nstat	Vector of print options for the nine statistics associated with each Beta or Delta coefficient. nstat(i) = 0 : No print for the ith statistic (i=1,9). i : Print the ith statistic. The nine statistics are: i = 1 : BETA or DELTA coefficient i = 2 : Derivative of Y (sample value) i = 3 : Derivative of E(Y) i = 4 : Elasticity of Y (sample value) i = 5 : Elasticity of E(Y) i = 6 : Correlation (Y,X_k or Z_m) original units i = 7 : Correlation (Y,X_k or Z_m) Box-Cox transformed i = 8 : Marginal Rate of Substitution (d X_ref / d X_k or Z_m) i = 9 : Conditional t-statistic for BETA or DELTA.

Second Card Type (Variables of Reference for MRS between two X's)	
xrefname	Vector of names of the reference variables X_ref for the Marginal Rate of Substitution (MRS) between the reference variable X_ref and each independent variable X_k (or heteroskedasticity variable Z_m), for the NEQ equations. Enter one name per line in Fortran format A8.
Third Card Type (Variables for Heteroskedasticity, necessary only if NDT>0)	
namz	Vector of NDT names of the variables Z used for heteroskedasticity in the NEQ equations. Enter one name per line in Fortran format A8.
SECTION II :List of Names and Definitions of the Blocks, the Variables and the contiguity matrices R	
fname	Name of the file containing the list of all the names and definitions of the blocks and the independent variables within each block. The last block is reserved for the definitions and names of the contiguity matrices R. A blank card is needed after each block. The file can be in the current directory where TABLEX is run or in another directory in which case the full pathname of the directory should be specified.
SECTION III : Estimation Results	
fname	Names of the files containing the estimation results of the NEQ equations which were stored by program LEVEL 2.1. Enter one file name per card. The display order of the equations (or variants) in the columns of L-2.1 TABLEX follows the entry order of the file names specified in this section.

2.2 Running L-2.1 Tablex

Since L-2.1 Tablex is a DOS-program, it can only be run in a DOS dialog box:

1. Select START/RUN menu and type the command **cmd** to open the DOS dialog box.
2. At the DOS prompt, change to the directory where the Tablex input file is stored.
3. To run the program, type the command:
   D:\Level2new\L2Tablex\Release\L2Tablex2 < input file name > output file name.

The output file containing a set of estimated variants are stored in the same directory as the input file. An easier way to run the program without the DOS dialog box is to store the command in a batch file (extension .bat) and double-click on this file to execute the command. We will use this method to produce a tablex for the two examples below.

2.3 First Example of Input File

The first example of a Tablex input file, *Tablex.in,* is stored in *d:\Level2new\Tablex\Tablex Test\First Example* where *d:* is the drive where the L-2.1 program is installed.

Input file: Tablex.in

```
$card nbloc=1,nmatr=1,neq=6,ndt=1,nvs=1,nlist=0,nstat=1,2,3,4,5,6,7,8,9$
DIST
DIST
DIST
DIST
DIST
DIST
DPMDPX
L2Listdef.txt
D:\Level2new\Level2Tests\First Example\L2Variant1.tab
D:\Level2new\Level2Tests\First Example\L2Variant2.tab
D:\Level2new\Level2Tests\First Example\L2Variant3.tab
D:\Level2new\Level2Tests\First Example\L2Variant4.tab
D:\Level2new\Level2Tests\First Example\L2Variant5.tab
D:\Level2new\Level2Tests\First Example\L2Variant6.tab
```

Two files are used in *Tablex.in* :

- *L2Listdef.txt* which contains the names and the definitions of the blocks and of the variables in each block. This file is as follows:

```
1IND   = VARIABLES INDÉPENDANTES            INDEPENDENT VARIABLES
1AGDPM   PRODUIT INTÉRIEUR BRUT (ORIGINE)   GROSS DOMESTIC PRODUCT AT ORIGIN
1AGDPX   PRODUIT INTÉRIEUR BRUT (DESTINAT.) GROSS DOMESTIC PRODUCT AT DESTINAT.
1DIST    DISTANCE                           DISTANCE
3CDUMMY  DUMMY COMMERCE INTERPROVINCIAL     INTERPROVINCIAL TRADE DUMMY
1DPMDPX  PRODUIT AGDPM*AGDPX                PRODUCT AGDPM*AGDPX
1CONSTANTCONSTANTE DE LA RÉGRESSION         REGRESSION CONSTANT

 MATRICES
 DIST_OD O et D: DIST(400 milles)           O and D: DIST(400 miles)
```

The Fortran input format for this file is (i2,a8,2a35):

1. The first field length is 2 characters. It is reserved for the number of a block or the category of each independent variable X .
2. The second field length is 8 characters. It is used for the name of a block or a variable. Note that the name of the regression constant should be always in big letters wheras the names of the other variables can be in big letters if they are entered as such in the database.
3. The length of the last two fields is 35 characters each. In the first field, the definition of a block or a variable is entered in French. The second field is used for the English definition.

A number is associated with each block and a category is assigned to each independent variable:

1. Category 0: strictly positive variable and not a dummy $(X > 0)$.
2. Category 1: not strictly positive and not a dummy $(-\infty < X < +\infty)$.
3. Category 2: quasi-dummy $(X \geq 0)$.
4. Category 3: real dummy (X has only two values: 0 and a positive constant).

The category determines the way in which the elasticity of the dependent variable with respect to X is computed. For categories 2 and 3, the elasticity is corrected for the positive values of the variable X only. For further details about the computation of the elasticity, see *Tran and Gaudry (2008).*

- *L2VariantX.tab* which contains the estimation results associated with the *X*-th *Variant.*

2.4 First Example of Output File

An example of output file, *Tablex.txt,* is stored in the same directory as the input file *Tablex.in*. It is obtained by running the batch file *RunTablex.bat.*

Output file: Tablex.txt

```
&CARD
 NBLOC   =           1,
 NMATR   =           1,
 NEQ     =           6,
 NDT     =           1,
 NVS     =           1,
 NLIST   =           0,
 NSTAT   =           1,           2,           3,           4,           5,           6,           7,           8,           9
 /



 ==========================================================================================================
 PART I. Presented statistics
   Model type                                          L-2.1     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1
   Name of the variant                                   LOG    LOG+AU LOG+AU+PR BC1+AU+PR BC2+AU+PR BC2GHAUPR
   Version number of the variant                           1         2         3         4         5         6
   Dependent variable in the variant                    SHIP      SHIP      SHIP      SHIP      SHIP      SHIP
   Reference variable X_ref for M.R.S.                  DIST      DIST      DIST      DIST      DIST      DIST

 ==========================================================================================================

 -----------------------------
 IND   = INDEPENDENT VARIABLES
 -----------------------------

   GROSS DOMESTIC PRODUCT AT ORIGIN     AGDPM
     BETA coefficient                              0.11D+01  0.11D+01  0.11D+01  0.16D+01  0.19D+01  0.15D+01
     Derivative of Y (sample value)                0.43D+04  0.44D+04  0.43D+04  0.38D+04  0.37D+04  0.41D+04
     Derivative of E(Y)                            0.13D+05  0.99D+04  0.10D+05  0.54D+04  0.55D+04  0.66D+04
     Elasticity of Y (sample value)                   1.062     1.084     1.055     0.952     0.925     1.019
     Elasticity of E(Y)                               1.062     1.084     1.055     1.017     1.016     1.073
     Correlation (Y,X_k) original units               0.169     0.169     0.169     0.169     0.169     0.169
     Correlation (Y,X_k) Box-Cox transf.              0.169     0.430     0.456     0.450     0.448     0.745
     Marg. Rate of Subst.(d X_ref/d X_k)          -0.11D+02 -0.11D+02 -0.10D+02 -0.99D+01 -0.10D+02 -0.12D+02
     Conditional t-statistic for BETA             (  34.46) (  26.48) (  25.58) (  26.76) (  26.88) (  25.82)
                                                     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1

   GROSS DOMESTIC PRODUCT AT DESTINAT.  AGDPX
     BETA coefficient                              0.12D+01  0.12D+01  0.12D+01  0.18D+01  0.21D+01  0.17D+01
     Derivative of Y (sample value)                0.49D+04  0.50D+04  0.49D+04  0.43D+04  0.42D+04  0.44D+04
     Derivative of E(Y)                            0.12D+05  0.10D+05  0.11D+05  0.56D+04  0.56D+04  0.67D+04
     Elasticity of Y (sample value)                   1.206     1.236     1.226     1.073     1.044     1.104
     Elasticity of E(Y)                               1.206     1.236     1.226     1.145     1.147     1.163
     Correlation (Y,X_k) original units               0.180     0.180     0.180     0.180     0.180     0.180
     Correlation (Y,X_k) Box-Cox transf.              0.180    -0.113    -0.088    -0.059    -0.063    -0.011
     Marg. Rate of Subst.(d X_ref/d X_k)          -0.13D+02 -0.13D+02 -0.12D+02 -0.11D+02 -0.11D+02 -0.13D+02
     Conditional t-statistic for BETA             (  38.73) (  28.34) (  28.45) (  29.19) (  29.27) (  28.11)
                                                     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1

   DISTANCE                             DIST
     BETA coefficient                             -0.14D+01 -0.15D+01 -0.15D+01 -0.21D+01 -0.26D+01 -0.19D+01
     Derivative of Y (sample value)               -0.38D+03 -0.39D+03 -0.41D+03 -0.39D+03 -0.37D+03 -0.35D+03
     Derivative of E(Y)                           -0.39D+04 -0.28D+04 -0.29D+04 -0.11D+04 -0.11D+04 -0.12D+04
     Elasticity of Y (sample value)                  -1.425    -1.458    -1.534    -1.446    -1.388    -1.297
     Elasticity of E(Y)                              -1.425    -1.458    -1.534    -1.613    -1.561    -1.378
     Correlation (Y,X_k) original units              -0.213    -0.213    -0.213    -0.213    -0.213    -0.213
     Correlation (Y,X_k) Box-Cox transf.             -0.213     0.111     0.141     0.086     0.085     0.181
     Marg. Rate of Subst.(d X_ref/d X_k)           0.10D+01  0.10D+01  0.10D+01  0.10D+01  0.10D+01  0.10D+01
     Conditional t-statistic for BETA             ( -22.75) ( -13.80) ( -13.75) ( -15.16) ( -15.36) ( -14.89)
                                                     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1

   INTERPROVINCIAL TRADE DUMMY          CDUMMY
```

```
  (Dummy)                            ======
    BETA coefficient                          0.31D+01  0.30D+01  0.30D+01  0.59D+01  0.61D+01  0.46D+01
    Derivative of Y (sample value)            0.11D+07  0.11D+07  0.11D+07  0.94D+06  0.94D+06  0.10D+07
    Derivative of E(Y)                        0.29D+07  0.20D+07  0.20D+07  0.85D+06  0.83D+06  0.10D+07
    Elasticity of Y (sample value)               3.093     2.980     3.024     2.621     2.617     2.848
    Elasticity of E(Y)                           3.093     2.980     3.024     2.708     2.709     2.861
    Correlation (Y,X_k) original units           0.172     0.172     0.172     0.172     0.172     0.172
    Correlation (Y,X_k) Box-Cox transf.          0.172    -0.108    -0.093    -0.098    -0.102    -0.056
    Marg. Rate of Subst.(d X_ref/d X_k)      -0.29D+04 -0.28D+04 -0.27D+04 -0.24D+04 -0.25D+04 -0.30D+04
    Conditional t-statistic for BETA         (  23.78) (  23.61) (  24.06) (  28.34) (  28.26) (  27.34)


  REGRESSION CONSTANT                CONSTANT
    BETA coefficient                          0.12D+02  0.12D+02  0.12D+02  0.18D+02  0.19D+02  0.14D+02
    Derivative of Y (sample value)            0.41D+07  0.42D+07  0.44D+07  0.28D+07  0.29D+07  0.31D+07
    Derivative of E(Y)                        0.11D+08  0.78D+07  0.82D+07  0.26D+07  0.26D+07  0.31D+07
    Elasticity of Y (sample value)              11.512    11.569    12.221     7.884     8.097     8.581
    Elasticity of E(Y)                          11.512    11.569    12.221     8.868     9.143     9.142
    Correlation (Y,X_k) original units           0.000     0.000     0.000     0.000     0.000     0.000
    Correlation (Y,X_k) Box-Cox transf.          0.000     0.000     0.000     0.000     0.000     0.000
    Marg. Rate of Subst.(d X_ref/d X_k)      -0.11D+05 -0.11D+05 -0.11D+05 -0.73D+04 -0.79D+04 -0.89D+04
    Conditional t-statistic for BETA         (  23.61) (  14.56) (  14.71) (  13.75) (  13.30) (  12.74)


Heteroskedasticity
------------------

  PRODUCT AGDPM*AGDPX                DPMDPX
    DELTA coefficient                                                                          -0.21D+00
    Derivative of Y (sample value)                                                             -0.52D+01
    Derivative of E(Y)                                                                         -0.27D+02
    Elasticity of Y (sample value)                                                                -0.093
    Elasticity of E(Y)                                                                            -0.085
    Correlation (Y,Z_m) original units                                                             0.398
    Correlation (Y,Z_m) Box-Cox transf.                                                            0.701
    Marg. Rate of Subst.(d X_ref/d Z_m)                                                         0.15D-01
    Conditional t-statistic for DELTA                                                          (  -7.06)

==========================================================================================================
PART II. Parameters
         t-statistic unconditional (=0) [=1]
  Model type                                     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1
  Name of the variant                              LOG    LOG+AU LOG+AU+PR BC1+AU+PR BC2+AU+PR BC2GHAUPR
  Version number of the variant                      1         2         3         4         5         6
  Dependent variable in the variant               SHIP      SHIP      SHIP      SHIP      SHIP      SHIP

==========================================================================================================

Box-Cox Transformations
-----------------------

  LAMBDA(Y)                                      0.000     0.000     0.000     0.064     0.066     0.037
                                                 Fixed     Fixed     Fixed (   8.34) (   8.52) (   4.00)
                                                                           [-122.15] [-120.90] [-104.05]

  LAMBDA(X) - Group 1                            0.000     0.000     0.000     0.064     0.032     0.014
                                                 Fixed     Fixed     Fixed (   8.34) (   1.35) (   0.52)
                                                                           [-122.15] [ -40.67] [ -36.34]

Heteroskedasticity
------------------

  PRODUCT AGDPM*AGDPX                DPMDPX

     LAMBDA(Z)                                                                                     0.000
                                                                                                   Fixed


Directed Autocorrelation
------------------------

  O and D: DIST(400 miles)           DIST_OD

     RHO                                                   0.530     0.662     0.670     0.651     0.656
                                                       (  10.88) (  11.46) (  11.19) (  10.46) (  10.58)

     PI                                                    1.000     0.383     0.387     0.422     0.459
                                                           Fixed (   3.12) (   2.65) (   2.65) (   2.95)
                                                                 [  -5.02] [  -4.19] [  -3.63] [  -3.48]

==========================================================================================================
PART III. General statistics
  Model type                                     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1
  Name of the variant                              LOG    LOG+AU LOG+AU+PR BC1+AU+PR BC2+AU+PR BC2GHAUPR
  Version number of the variant                      1         2         3         4         5         6
  Dependent variable in the variant               SHIP      SHIP      SHIP      SHIP      SHIP      SHIP

==========================================================================================================

Log-likelihood                                -8089.56  -8019.24  -8011.54  -7976.36  -7975.42  -7960.31

Degrees of freedom                                   4         5         6         7         8         9

Pearson R2                                       0.514     0.582     0.575     0.684     0.683     0.634
Pearson R2 adjusted for Degrees of freedom       0.511     0.578     0.572     0.681     0.679     0.630
```

```
Pseudo-(L)-R2                                     0.812     0.779     0.794     0.799     0.799     0.906
Pseudo-(L)-R2 adjusted for Degrees of freedom     0.811     0.777     0.792     0.797     0.796     0.905

Average probability (Y=limit observation)         0.000     0.000     0.000     0.000     0.000     0.000

Sample     - Number of observations                 683       683       683       683       683       683

Total number of fixed or estimated parameters:
           - Beta .Estimated                          4         4         4         4         4         4
                  .Constant                           1         1         1         1         1         1
           - Lambda(y)
                  .Fixed                              1         1         1         0         0         0
                  .Estimated .Constrained             0         0         0         1         0         0
                             .Distinct                0         0         0         0         1         1
           - Lambda(x)
                  .Fixed                              1         1         1         0         0         0
                  .Estimated                          0         0         0         1         1         1
           - Heteroskedasticity
              Delta
                  .Fixed                              0         0         0         0         0         0
                  .Estimated                          0         0         0         0         0         1
              Lambda(z)
                  .Fixed                              0         0         0         0         0         1
                  .Estimated                          0         0         0         0         0         0
           - Autocorrelation
              Rho .Estimated                          0         1         1         1         1         1
              Pi  .Fixed                              0         1         0         0         0         0
                  .Estimated                          0         0         1         1         1         1
           - Total
                  .Fixed                              2         3         2         0         0         1
                  .Estimated                          5         6         7         8         9        10

Mean of observed Y                             0.36D+06  0.36D+06  0.36D+06  0.36D+06  0.36D+06  0.36D+06

Mean of estimated E(Y)                         0.94D+06  0.67D+06  0.67D+06  0.36D+06  0.36D+06  0.38D+06

=========================================================================================================
```

2.5 Second Example of Input File

The second example of a Tablex input file, *Tablex.in*, is stored in *d:\Level2new\Tablex\Tablex Test\Second Example* where *d:* is the drive where the L-2.1 program is installed.

Input file: Tablex.in

```
$card nbloc=1,nmatr=1,neq=6,ndt=0,nvs=1,nlist=0,nstat=1,2,3,4,5,6,7,8,9$
REVE
REVE
REVE
REVE
REVE
REVE
L2Listdef.txt
D:\Level2new\Level2Tests\Second Example\L2Variant1.tab
D:\Level2new\Level2Tests\Second Example\L2Variant2.tab
D:\Level2new\Level2Tests\Second Example\L2Variant3.tab
D:\Level2new\Level2Tests\Second Example\L2Variant4.tab
D:\Level2new\Level2Tests\Second Example\L2Variant5.tab
D:\Level2new\Level2Tests\Second Example\L2Variant6.tab
```

Two files are used in *Tablex.in*:

- *L2Listdef.txt* which contains the names and the definitions of the blocks and of the variables in each block. This file is as follows:

```
1IND   = VARIABLES INDÉPENDANTES           INDEPENDENT VARIABLES
0POP     POPULATION                        POPULATION
0REVE    REVENU                            REVENUE
0LANGFR  LANGUE FRANÇAISE                  FRENCH LANGUAGE
0UTILITY UTILITÉ                           UTILITY
0CONSTANTCONSTANTE DE LA RÉGRESSION        REGRESSION CONSTANT
```

```
MATRICES
DIST_OD O et D: DIST(320km)                O and D: DIST(320km)
```

- *L2VariantX.tab* which contains the estimation results associated with the *X*-th *Variant*.

2.6 Second Example of Output File

The second example of output file, *Tablex.txt,* is stored in the same directory as the input file *Tablex.in*. It is obtained by running the batch file *RunTablex.bat.*

Output file: Tablex.txt

```
&CARD
 NBLOC   =           1,
 NMATR   =           1,
 NEQ     =           6,
 NDT     =           0,
 NVS     =           1,
 NLIST   =           0,
 NSTAT   =           1,           2,           3,           4,           5,           6,           7,           8,           9
 /



 ==============================================================================================================
 PART I. Presented statistics
   Model type                                          L-2.1     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1
   Name of the variant                                   LOG    LOG+AU LOG+AU+PR BC1+AU+PR BC2+AU+PR BC5+AU+PR
   Version number of the variant                           1         2         3         4         5         6
   Dependent variable in the variant                 TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW
   Reference variable X_ref for M.R.S.                  REVE      REVE      REVE      REVE      REVE      REVE

 ==============================================================================================================

 -----------------------------
 IND   = INDEPENDENT VARIABLES
 -----------------------------

   POPULATION                          POP
     BETA coefficient                              0.14D+01  0.14D+01  0.14D+01  0.14D+01  0.51D+01  0.38D+00
     Derivative of Y (sample value)                0.39D+00  0.40D+00  0.40D+00  0.39D+00  0.38D+00  0.41D+00
     Derivative of E(Y)                            0.50D+00  0.45D+00  0.45D+00  0.42D+00  0.40D+00  0.39D+00
     Elasticity of Y (sample value)                   1.389     1.419     1.425     1.410     1.378     1.474
     Elasticity of E(Y)                               1.389     1.419     1.425     1.441     1.506     1.540
     Correlation (Y,X_k) original units               0.412     0.412     0.412     0.412     0.412     0.412
     Correlation (Y,X_k) Box-Cox transf.              0.412     0.900     0.905     0.894     0.795     0.851
     Marg. Rate of Subst.(d X_ref/d X_k)           0.13D-01  0.63D-02  0.67D-02  0.62D-02  0.99D-02  0.19D-01
     Conditional t-statistic for BETA            (  25.45) (  28.04) (  28.36) (  29.98) (  32.66) (  37.50)
                                                    LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 1

   REVENUE                             REVE
     BETA coefficient                              0.83D+00  0.18D+01  0.17D+01  0.20D+01  0.33D+01  0.34D-72
     Derivative of Y (sample value)                0.29D+02  0.62D+02  0.59D+02  0.64D+02  0.39D+02  0.22D+02
     Derivative of E(Y)                            0.36D+02  0.73D+02  0.69D+02  0.68D+02  0.37D+02  0.32D+02
     Elasticity of Y (sample value)                   0.832     1.799     1.706     1.840     1.123     0.633
     Elasticity of E(Y)                               0.832     1.799     1.706     1.891     1.211     1.138
     Correlation (Y,X_k) original units               0.057     0.057     0.057     0.057     0.057     0.057
     Correlation (Y,X_k) Box-Cox transf.              0.057     0.818     0.828     0.800     0.717     0.293
     Marg. Rate of Subst.(d X_ref/d X_k)           0.10D+01  0.10D+01  0.10D+01  0.10D+01  0.10D+01  0.10D+01
     Conditional t-statistic for BETA            (   1.20) (   2.39) (   2.22) (   2.50) (   1.65) (   2.23)
                                                    LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 2

   FRENCH LANGUAGE                     LANGFR
     BETA coefficient                              0.66D-01  0.33D+00  0.35D+00  0.42D+00  0.72D+00  0.77D+00
     Derivative of Y (sample value)                0.13D+03  0.67D+03  0.70D+03  0.72D+03  0.59D+03  0.39D+03
     Derivative of E(Y)                            0.19D+03  0.91D+03  0.95D+03  0.91D+03  0.70D+03  0.52D+03
     Elasticity of Y (sample value)                   0.066     0.334     0.348     0.358     0.291     0.194
     Elasticity of E(Y)                               0.066     0.334     0.348     0.366     0.317     0.216
     Correlation (Y,X_k) original units               0.030     0.030     0.030     0.030     0.030     0.030
     Correlation (Y,X_k) Box-Cox transf.              0.030     0.786     0.796     0.774     0.701     0.693
     Marg. Rate of Subst.(d X_ref/d X_k)           0.46D+01  0.11D+02  0.12D+02  0.11D+02  0.15D+02  0.18D+02
     Conditional t-statistic for BETA            (   1.20) (   4.08) (   4.16) (   4.42) (   4.69) (   6.69)
                                                    LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 3

   UTILITY                             UTILITY
     BETA coefficient                              0.63D+00  0.57D+00  0.57D+00  0.78D+00  0.91D+00  0.80D+00
     Derivative of Y (sample value)                0.19D+07  0.17D+07  0.17D+07  0.17D+07  0.16D+07  0.15D+07
     Derivative of E(Y)                            0.28D+08  0.26D+08  0.26D+08  0.25D+08  0.33D+08  0.39D+08
     Elasticity of Y (sample value)                   0.634     0.571     0.569     0.567     0.521     0.486
     Elasticity of E(Y)                               0.634     0.571     0.569     0.545     0.656     0.687
     Correlation (Y,X_k) original units               0.491     0.491     0.491     0.491     0.491     0.491
     Correlation (Y,X_k) Box-Cox transf.              0.491    -0.439    -0.463    -0.441    -0.307    -0.220
```

```
    Marg. Rate of Subst.(d X_ref/d X_k)        0.66D+05  0.27D+05  0.29D+05  0.27D+05  0.40D+05  0.67D+05
    Conditional t-statistic for BETA          (  42.33) (  25.07) (  25.27) (  25.99) (  21.13) (  24.37)
                                                 LAM 1     LAM 1     LAM 1     LAM 1     LAM 1     LAM 4

  REGRESSION CONSTANT                 CONSTANT
    BETA coefficient                          -0.11D+02 -0.21D+02 -0.20D+02 -0.24D+02 -0.54D+02  0.33D+01
    Derivative of Y (sample value)            -0.16D+07 -0.30D+07 -0.30D+07 -0.28D+07 -0.40D+07  0.24D+06
    Derivative of E(Y)                        -0.20D+07 -0.36D+07 -0.35D+07 -0.30D+07 -0.37D+07  0.22D+06
    Elasticity of Y (sample value)              -10.784   -20.650   -20.011   -18.640   -26.839     1.600
    Elasticity of E(Y)                          -10.784   -20.650   -20.011   -19.150   -28.952     1.733
    Correlation (Y,X_k) original units            0.000     0.000     0.000     0.000     0.000     0.000
    Correlation (Y,X_k) Box-Cox transf.           0.000     0.000     0.000     0.000     0.000     0.000
    Marg. Rate of Subst.(d X_ref/d X_k)       -0.55D+05 -0.49D+05 -0.50D+05 -0.43D+05 -0.10D+06  0.11D+05
    Conditional t-statistic for BETA          (  -1.96) (  -3.35) (  -3.19) (  -3.39) (  -4.02) (   3.81)


==========================================================================================================
PART II. Parameters
         t-statistic unconditional (=0) [=1]
  Model type                                      L-2.1     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1
  Name of the variant                               LOG    LOG+AU LOG+AU+PR BC1+AU+PR BC2+AU+PR BC5+AU+PR
  Version number of the variant                       1         2         3         4         5         6
  Dependent variable in the variant             TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW

==========================================================================================================

Box-Cox Transformations
-----------------------

  LAMBDA(Y)                                       0.000     0.000     0.000     0.021     0.059     0.062
                                                  Fixed     Fixed     Fixed (   1.39) (   4.02) (   2.40)
                                                                          [ -63.10] [ -64.36] [ -36.42]

  LAMBDA(X) - Group 1                             0.000     0.000     0.000     0.021    -0.047     0.158
                                                  Fixed     Fixed     Fixed (   1.39) (  -1.96) (   2.57)
                                                                          [ -63.10] [ -43.92] [ -13.70]

  LAMBDA(X) - Group 2                                                                              20.000
                                                                                              (   1.24)
                                                                                              [   1.18]

  LAMBDA(X) - Group 3                                                                              -0.149
                                                                                              (  -0.43)
                                                                                              [  -3.31]

  LAMBDA(X) - Group 4                                                                              -0.078
                                                                                              (  -3.24)
                                                                                              [ -44.72]

Directed Autocorrelation
------------------------

  O and D: DIST(320km)                DIST_OD

     RHO                                                    0.747     0.762     0.782     0.768     0.860
                                                        (  11.00) (  10.95) (  12.63) (  10.71) (   6.81)

     PI                                                     1.000     0.841     0.860     0.440     0.000
                                                            Fixed (   2.72) (   2.76) (   1.37) (   0.00)
                                                                  [  -0.52] [  -0.45] [  -1.74] [-37867.]

==========================================================================================================
PART III. General statistics
  Model type                                      L-2.1     L-2.1     L-2.1     L-2.1     L-2.1     L-2.1
  Name of the variant                               LOG    LOG+AU LOG+AU+PR BC1+AU+PR BC2+AU+PR BC5+AU+PR
  Version number of the variant                       1         2         3         4         5         6
  Dependent variable in the variant             TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW   TOTFLOW

==========================================================================================================

Log-likelihood                                 -1318.23  -1294.58  -1294.48  -1293.94  -1279.71  -1262.98

Degrees of freedom                                    4         5         6         7         8        11

Pearson R2                                        0.885     0.943     0.943     0.951     0.971     0.996
Pearson R2 adjusted for Degrees of freedom        0.881     0.941     0.940     0.948     0.969     0.996

Pseudo-(L)-R2                                     0.944     0.980     0.981     0.979     0.980     0.989
Pseudo-(L)-R2 adjusted for Degrees of freedom     0.942     0.979     0.980     0.978     0.978     0.988

Average probability (Y=limit observation)         0.000     0.000     0.000     0.000     0.000     0.000

Sample     - Number of observations                 120       120       120       120       120       120

Total number of fixed or estimated parameters:
           - Beta .Estimated                          4         4         4         4         4         4
                  .Constant                           1         1         1         1         1         1
           - Lambda(y)
                  .Fixed                              1         1         1         0         0         0
                  .Estimated .Constrained             0         0         0         1         0         0
                             .Distinct                0         0         0         0         1         1
           - Lambda(x)
                  .Fixed                              1         1         1         0         0         0
                  .Estimated                          0         0         0         1         1         4
           - Heteroskedasticity
```

```
              Delta
                  .Fixed                                       0         0         0         0         0         0
                  .Estimated                                   0         0         0         0         0         0
              Lambda(z)
                  .Fixed                                       0         0         0         0         0         0
                  .Estimated                                   0         0         0         0         0         0
           - Autocorrelation
              Rho .Estimated                                   0         1         1         1         1         1
              Pi  .Fixed                                       0         1         0         0         0         0
                  .Estimated                                   0         0         1         1         1         1
           - Total
                  .Fixed                                       2         3         2         0         0         0
                  .Estimated                                   5         6         7         8         9        12

Mean of observed Y                                 0.15D+06  0.15D+06  0.15D+06  0.15D+06  0.15D+06  0.15D+06

Mean of estimated E(Y)                             0.19D+06  0.17D+06  0.17D+06  0.16D+06  0.15D+06  0.15D+06

============================================================================================================
```

3. PROGRAM LISTINGS

3.1 Program LEVEL L-2.1

The program contains 30 modules listed below in alphabetical order:

```
array.f       correl.f      cputime.f     derzl.f       diffir.f      eigen.f
expelast.f    fcn.f         gmadd.f       gmprd.f       gmsub.f       gtprd.f
level2.f      loc.f         mata.f        mcpy.f        mfsd.f        minv.f
mprd.f        ndtr.f        pmatrix.f     shminv.f      sinv.f        smpelast.f
smpy.f        stormat.f     trace.f       vardec.f      vp.f          zlam.f
```

It was compiled with Compaq Visual Fortran 6.6. The main module *level2.f* contains an algorithm to find the maximum of the log-likehood function. This module calls the module *fcn.f* which computes the function and its first derivatives with respect to the estimated parameters. All the other modules used for matrix operations and different statistics computation are mainly called by *fcn.f*.

Program listing

```
      subroutine array(mode,i,j,n,m,s,d)
c     ..................................................................
c
c        subroutine array
c
c        purpose
c           convert data array from single to double dimension or vice
c           versa.  this subroutine is used to link the user program
c           which has double dimension arrays and the ssp subroutines
c           which operate on arrays of data in a vector fashion.
c
c        usage
c           call array (mode,i,j,n,m,s,d)
c
c        description of parameters
c           mode - code indicating type of conversion
c                    1 - from single to double dimension
c                    2 - from double to single dimension
c           i    - number of rows in actual data matrix
c           j    - number of columns in actual data matrix
c           n    - number of rows specified for the matrix d in
c                  dimension statement
c           m    - number of columns specified for the matrix d in
c                  dimension statement
c           s    - if mode=1, this vector is input which contains the
c                  elements of a data matrix of size i by j. column i+1
c                  of data matrix follows column i, etc. if mode=2,
c                  this vector is output representing a data matrix of
c                  size i by j containing its columns consecutively.
c                  the length of s is ij, where ij=i*j.
c           d    - if mode=1, this matrix of size n by m is output,
c                  containing a data matrix of size i by j in the first
c                  i rows and j columns. if mode=2, this n by m matrix
c                  is input containing a data matrix of size i by j in
c                  the first i rows and j columns.
```

```
c
c        remarks
c           vector s can be in the same location as matrix d.  vector s
c           is referred as a matrix in other ssp routines, since it
c           contains a data matrix.
c           this subroutine converts only general data matrices (storage
c           mode of 0).
c
c        subroutines and function subroutines required
c           none
c
c        method
c           refer to the discussion on variable data size in the section
c           describing overall rules for usage in this manual.
c
c     ..................................................................
c
      implicit double precision (a-h,o-z)
      dimension s(1),d(1)
c
      ni=n-i
c
c        test type of conversion
c
      if(mode-1) 100, 100, 120
c
c        convert from single to double dimension
c
  100 ij=i*j+1
      nm=n*j+1
      do 110 k=1,j
      nm=nm-ni
      do 110 l=1,i
      ij=ij-1
      nm=nm-1
  110 d(nm)=s(ij)
      go to 140
c
c        convert from double to single dimension
c
  120 ij=0
      nm=0
      do 130 k=1,j
      do 125 l=1,i
      ij=ij+1
      nm=nm+1
  125 s(ij)=d(nm)
  130 nm=nm+ni
c
  140 return
      end

      subroutine correl(x,xm,s,cor,namx,namy,n,nobs,nb,nxx,ncor)
c INPUT:
c   X     (n x nb) Matrix of (original or transformed) Independent Variables,
c         excluding the Constant which has been replaced by the (original or
c         transformed) Dependent Variable Y, used in ESTIMATION.
```

```
c   XM    (nb) Vector of Means of the Independent Variables, excluding the
c         Constant which has been replaced by the Dependent Variable Y, used
c         in ESTIMATION.
c   NAMX  (nb) Vector of Names of the Independent Variables, including the
c         Constant, used in ESTIMATION.
c   NAMY  Name of the Dependent Variable Y used in ESTIMATION.
c   N     Maximum Number of Observations as declared on the Parameter card
c         in Subroutine FCN.
c   NOBS  Number of Observations used in ESTIMATION.
c   NB    Number of Independent Variables, including the Constant, used in
c         ESTIMATION.
c   NXX   Maximum Number of Different Elements in the (nx x nx) Symmetric
c         Matrix COR, stored columnwise as an upper triangular matrix.
c         This number is declared on the Parameter card in Subroutine FCN.

c   NCOR  = 2 - Table of Variance-Decomposition Proportions for the independent

c               variables X (including the Constant) and Correlation Matrix for

c               the independent variables X (excluding the Constant) and the

c               dependent variable Y. Both tables are computed and printed for

c               the original and transformed variables.

c         = 1 - Both tables are computed and printed for the transformed varia-

c               bles only.

c         = 0 - No print for the two tables, but only the correlation matrix

c               for X and Y is computed for TABLEX program which needs the

c               results in terms of the original and transformed variables.
c WORK VECTOR:
c   S     (nb) Vector used to store temporarily the diagonal elements of the
c         matrix of cross-products X'X where X is expressed in deviations
c         around the mean XM.
c OUTPUT:
c   COR   (nx x nx) Symmetric Matrix, stored columnwise as an upper triangular
c         matrix containing the pairwise correlations among the Independent
c         Variables X (excluding the Constant) and the Dependent Variable Y.
c PURPOSE:
c   Compute the Correlation Matrix of the Independent Variables X (excluding
c   the Constant) and the Dependent Variable Y.
c   The correlation matrix is stored columnwise in an upper triangular form,
c   but is printed in a lower triangular form.

      implicit double precision (a-h,o-z)

      real*4 epsi,wght

      character*10 namx(nb),namy
      dimension x(n,nb),xm(nb),s(nb),cor(nxx)

      common nlist,nstep,epsi,wght,noprint
```

```
      data ncol/12/


c April 20 2001: always compute the correlation matrix 'cor' even if ncor=0,

c                because TABLEX program needs the results for the correlation

c                between Y and every X.
      ij=0
      ir=0
      do 1 j=1,nb
      do 1 i=1,j
      ir=ir+1
      cor(ir)=-nobs*xm(i)*xm(j)
      do 2 k=1,nobs
    2 cor(ir)=cor(ir)+x(k,i)*x(k,j)
      if(i.ne.j)go to 1
      ij=ij+1
      s(ij)=cor(ir)
    1 continue
      ir=0
      do 3 j=1,nb
      do 3 i=1,j
      ir=ir+1
    3 cor(ir)=cor(ir)/dsqrt(s(i)*s(j))



c April 20 2001: print only if ncor =1 or ncor=2

      if(ncor.eq.1.or.ncor.eq.2)then
         namx(nb)=namy
         np=nb/ncol
         if(nb.gt.np*ncol)np=np+1
         do 5 i=1,np
         if(noprint.eq.0)print 4
    4    format(///,'correlation matrix'//)
         i1=1+(i-1)*ncol
         i2=i*ncol
         if(nb.lt.i2)i2=nb
         do 7 j=i1,nb
         call loc(j,i1,j1,nb,nb,1)
         j2=(j*(j+1))/2
         if(j.gt.i2)call loc(j,i2,j2,nb,nb,1)
    7    if(noprint.eq.0)print 6,namx(j),(cor(k),k=j1,j2)
    6    format(/a10,12f10.4)
    5    if(noprint.eq.0)print 8,(namx(k),k=i1,i2)
    8    format(//12x,12a10)
         namx(nb)='CONSTANT'

      endif


      return
      end
```

```
      double precision function derzl(z,vl)
c   compute the derivative of the box-cox transformation z(lambda)
c   with respect to lambda, where vl is a value of lambda.
      implicit double precision (a-h,o-z)
      derzl=0.
      if(z.eq.0.)return
      if(dabs(vl).ge.1.d-4)go to 1
      derzl=0.5*dlog(z)**2
      return
    1 derzl=((z**vl)*dlog(z)-zlam(z,vl))/vl
      return
      end

      subroutine diffir(r,d,n)
c******************************************
c Compute an (N x N) matrix D = I - R
c where I is an identity matrix of order N,
c and   R is an (N x N) general matrix.
c******************************************
      implicit double precision (a-h,o-z)
      dimension r(*),d(*)
      do 1 i=1,n
      do 2 j=1,n
      ij=i+(j-1)*n
2     d(ij)=-r(ij)
      id=i+(i-1)*n
1     d(id)=1.d0 + d(id)
      return
      end

      subroutine eigen(a,r,n,mv)
c  a  = original matrix (symmetric), destroyed in computation. Resultant
c       eigenvalues are developed in diagonal of matrix (a) in descending
c       order.
c  r  = resultant matrix of eigenvectors (stored columnwise, in same
c       sequence as eigenvalues).
c  n  = order of matrices (a) and (r).
c  mv = 0  compute eigenvalues and eigenvectors.
c     = 1  compute eigenvalues only ((r) need not be dimensioned but must
c          still appear in calling sequence).
      implicit double precision(a-h,o-z)
      dimension a(*),r(*)
      if(mv-1)10,25,10
   10 iq=-n
      do 20 j=1,n
      iq=iq+n
      do 20 i=1,n
      ij=iq+i
      r(ij)=0.d0
      if(i-j)20,15,20
   15 r(ij)=1.d0
   20 continue
   25 anorm=0.d0
      do 35 i=1,n
      do 35 j=i,n
      if(i-j)30,35,30
   30 ia=i+(j*j-j)/2
```

```
      anorm=anorm+a(ia)*a(ia)
   35 continue
      if(anorm)165,165,40
   40 anorm=dsqrt(2*anorm)
      anrmx=anorm*1.d-6/dble(n)
      ind=0
      thr=anorm
   45 thr=thr/dble(n)
   50 l=1
   55 m=l+1
   60 mq=(m*m-m)/2
      lq=(l*l-l)/2
      lm=l+mq
      if(dabs(a(lm))-thr)130,65,65
   65 ind=1
      ll=l+lq
      mm=m+mq
      x=0.5d0*(a(ll)-a(mm))
      y=-a(lm)/dsqrt(a(lm)*a(lm)+x*x)
      if(x)70,75,75
   70 y=-y
   75 sinx=y/dsqrt(2*(1+(dsqrt(1-y*y))))
      sinx2=sinx*sinx
      cosx=dsqrt(1-sinx2)
      cosx2=cosx*cosx
      sincs=sinx*cosx
      ilq=n*(l-1)
      imq=n*(m-1)
      do 125 i=1,n
      iq=(i*i-i)/2
      if(i-l)80,115,80
   80 if(i-m)85,115,90
   85 im=i+mq
      go to 95
   90 im=m+iq
   95 if(i-l)100,105,105
  100 il=i+lq
      go to 110
  105 il=l+iq
  110 x=a(il)*cosx-a(im)*sinx
      a(im)=a(il)*sinx+a(im)*cosx
      a(il)=x
  115 if(mv-1)120,125,120
  120 ilr=ilq+i
      imr=imq+i
      x=r(ilr)*cosx-r(imr)*sinx
      r(imr)=r(ilr)*sinx+r(imr)*cosx
      r(ilr)=x
  125 continue
      x=2*a(lm)*sincs
      y=a(ll)*cosx2+a(mm)*sinx2-x
      x=a(ll)*sinx2+a(mm)*cosx2+x
      a(lm)=(a(ll)-a(mm))*sincs+a(lm)*(cosx2-sinx2)
      a(ll)=y
      a(mm)=x
  130 if(m-n)135,140,135
  135 m=m+1
```

```
      go to 60
  140 if(l-(n-1))145,150,145
  145 l=l+1
      go to 55
  150 if(ind-1)160,155,160
  155 ind=0
      go to 50
  160 if(thr-anrmx)165,165,45
c SORT EIGENVALUES AND EIGENVECTORS BY INCREASING VALUES
  165 iq=-n
      do 185 i=1,n
      iq=iq+n
      ll=i+(i*i-i)/2
      jq=n*(i-2)
      do 185 j=i,n
      jq=jq+n
      mm=j+(j*j-j)/2
      if(a(mm)-a(ll))170,185,185
  170 x=a(ll)
      a(ll)=a(mm)
      a(mm)=x
      if(mv-1)175,185,175
  175 do 180 k=1,n
      ilr=iq+k
      imr=jq+k
      x=r(ilr)
      r(ilr)=r(imr)
  180 r(imr)=x
  185 continue
      return
      end

      subroutine expelast(y,x,xbox,z,p,v,sqrtfz,beta,delta,vly,vlx,vlz,
     +sigw2,n,nx,nz,nobs,nullx,nb,nvaz,nlamy,numlx,numlz,numxeqz,
     +expy,expderx,expelax,expderz,expelaz,ysam,r2e,plim,eymean,nexpy,
     +iflag)
c Version with the formulas involving P and V.
c Input arguments : (y,...,numxeqz) (see Definitions in Subroutine FCN).
c Output arguments: (expy,...,eymean) (see Definitions below).
      implicit double precision (a-h,o-z)
      dimension y(n),x(n,nx),xbox(n,nx),z(n,nz)
      dimension p(*),v(*),sqrtfz(*),beta(*),delta(*)
      dimension vlx(*),vlz(*),nullx(*)
      dimension numlx(*),numlz(*),numxeqz(*),ysam(*)
      dimension expy(*),expderx(*),expelax(*),expderz(*),expelaz(*)
      dimension tgauss(32),wgauss(32)
      data sq2pi/2.5066282746310d0/
      data (tgauss(i),i=17,32)/.48307665687738d-1,.14447196158280d0,
     +.23928736225214d0,.33186860228213d0,.42135127613064d0,
     +.50689990893223d0,.58771575724076d0,.66304426693022d0,
     +.73218211874029d0,.79448379596794d0,.84936761373257d0,
     +.89632115576605d0,.93490607593774d0,.96476225558751d0,
     +.98561151154527d0,.99726386184948d0/
      data (wgauss(i),i=17,32)/.96540088514728d-1,.95638720079275d-1,
     +.93844399080805d-1,.91173878695764d-1,.87652093004404d-1,
     +.83311924226947d-1,.78193895787070d-1,.72345794108849d-1,
     +.65822222776362d-1,.58684093478536d-1,.50998059262376d-1,
```

```
     +.428358980222227d-1,.342738629130214d-1,.253920653092621d-1,
     +.162743947309057d-1,.701861000947010d-2/
c Generate the abscissas (tgauss) and weights (wgauss) of the Gauss-Legendre
c 32-points quadrature formula for the first 16 points.
      do 1 i=1,16
      j=33-i
      tgauss(i)=-tgauss(j)
1     wgauss(i)=wgauss(j)
c Compute the default value of the upper limit of Y if vly < 0: <uplimy>.
      if(nlamy.eq.1)then
         avly=dabs(vly)
         if(vly.lt.0.d0.and.avly.ge.1.d-4)then
            ymax=-1.d300
            ymean=0.d0
            do 2 i=1,nobs
            if(y(i).gt.ymax)ymax=y(i)
2           ymean=ymean+y(i)/nobs
            sdy2=0.d0
            do 3 i=1,nobs
3           sdy2=sdy2+(y(i)-ymean)**2
            sigy=dsqrt(sdy2/(nobs-1))
            uplimy=ymax+10*sigy
         endif
      endif

c Compute for each observation in the sample:
c    expy    : E(Y) = expected value of Y,
c    derx    : derivative of E(Y) with respect to X,
c    elax    : elasticity of E(Y) with respect to X,
c    derz    : derivative of E(Y) with respect to Z,
c    elaz    : elasticity of E(Y) with respect to Z.
c Compute over NOBS observations in the sample:
c    expderx : average value of the derivatives <derx>,
c    expelax : average value of the elasticities <elax>,
c    expderz : average value of the derivatives <derz>,
c    expelaz : average value of the elasticities <elaz>.
      sigw=dsqrt(sigw2)
      sqsw=sq2pi*sigw
      do 11 k=1,nb
      expderx(k)=0.d0
      expelax(k)=0.d0
11    continue
      if(nvaz.gt.0)then
         do 12 k=1,nvaz
         expderz(k)=0.d0
12       expelaz(k)=0.d0
      endif

      ymean=0.d0
      eymean=0.d0
      plim=0.d0
      if(nexpy.gt.0)then
      if(iflag.eq.1)then
            print 1405
1405        format(//'INITIAL STEP'/)
      else
            print 1407
```

```
1407        format(//'FINAL STEP'/)
      endif
            print 17
17          format('OBS.NO.',2x,'OBS.Y',10x,'EXP(Y)',7x,'OBS.Y - EXP(Y)',
     +      3x,'SAMPLE Y')
      endif

      do 15 i=1,nobs
      xb=0.d0
      do 4 k=1,nb
4     xb=xb+xbox(i,k)*beta(k)
      pv=0.d0
      do 5 k=1,nobs
      ik=i+(k-1)*nobs
      if(k.ne.i)then
         pv=pv+p(ik)*v(k)
      else
         psf=p(ik)
         if(nvaz.gt.0)psf=psf/sqrtfz(i)
      endif
5     continue
c Jan 28 2014: Compute sample Y
      if(avly.ge.1.d-4)then
            ysam(i)=(1+vly*xb)**(1/vly)
      else
            ysam(i)=dexp(xb)
      endif
c Default values of upper (bsup) and lower (binf) bounds of integration for W.
      bsup=5*sigw
      binf=-bsup
c By definition, PROB = 0: probability of Y to be at the limit = 0 if vly = 0.
      prob=0.d0
      if(nlamy.eq.1)then
c Case with LAMBDA(Y) different from zero.
         if(avly.ge.1.d-4)then
            if(vly.lt.0.d0)then
c Compute the effective upper bound of W if vly < 0.
               wupper=psf*(zlam(uplimy,vly)-xb)+pv
               if(wupper.lt.bsup)bsup=wupper
               bus=bsup/sigw
               call ndtr(bus,prob,dens)
               prob=1 - prob
            else
c Compute the effective lower bound of W if vly > 0.
               wlower=psf*((-1/vly)-xb)+pv
               if(wlower.gt.binf)binf=wlower
               bis=binf/sigw
               call ndtr(bis,prob,dens)
            endif
         endif
      else
c Case without LAMBDA(Y): compute the effective lower bound of W.
         wlower=-psf*xb + pv
         if(wlower.gt.binf)binf=wlower
         bis=binf/sigw
         call ndtr(bis,prob,dens)
      endif
```

```
c Compute PLIM : mean probability of Y to be at the limit.
      plim=plim+prob/nobs

c Compute the integrals with 32-points quadrature formula for:
c    expy    : E(Y) = expected value of Y,
c    factorx : factor common to all the derivatives and the elasticities of E(Y)
c              with respect to any independent variable X,
c    factorz : factor common to all the derivatives and the elasticities of E(Y)
c              with respect to any heteroskedasticity variable Z.
      rjac=(bsup-binf)/2
      const=(bsup+binf)/2
      expy(i)=0.d0
      factorx=0.d0
      factorz=0.d0
      do 6 j=1,32
      w=rjac*tgauss(j)+const
      fct=dexp(-0.5*w*w/sigw2)*wgauss(j)*rjac/sqsw
      wpv=(w-pv)/psf
      xbw=xb+wpv
      if(nlamy.eq.1)then
c Case with LAMBDA(Y).
         if(avly.ge.1.d-4)then
            yt=(1+vly*xbw)**(1/vly)
         else
            yt=dexp(xbw)
         endif
         ytpower=yt**(1-vly)
      else
c Case without LAMBDA(Y).
         yt=xbw
         ytpower=1.d0
      endif
      expy(i)=expy(i)+yt*fct
      factorx=factorx+ytpower*fct
      if(nvaz.gt.0)factorz=factorz+ytpower*fct*wpv
6     continue

c Add a second term to the expected value of Y if vly < 0.
      if(nlamy.eq.1.and.vly.lt.0.d0.and.avly.ge.1.d-4)then
         bss=bsup/sigw
         call ndtr(bss,prob,dens)
         expy(i)=expy(i)+uplimy*(1-prob)
      endif
c Compute expy(i)
      if(nexpy.gt.0)then
            print 166,i,y(i),expy(i),y(i)-expy(i),ysam(i)
166         format(i5,4d15.7)
      endif

c Compute <derx> and <elax> for each observation in the sample.
c Compute <expderx> and <expelax>.
      do 7 k=1,nb
      ngroup=numlx(k)
      if(ngroup.eq.0)then
c Case without LAMBDA(X).
         derx=factorx*beta(k)
      else
```

```
c Case with LAMBDA(X).
            derx=0.d0
            if(x(i,k).gt.0.d0)derx=factorx*beta(k)*x(i,k)**(vlx(ngroup)-1)
         endif
         elax=derx*x(i,k)/expy(i)
         expderx(k)=expderx(k)+derx/nobs
         if(nullx(k).eq.0)then
            expelax(k)=expelax(k)+elax/nobs
         else
c Correction of the elasticity of E(Y) with respect to a dummy or a quasi-dummy
c for the portion of positive observations of the variable.
            expelax(k)=expelax(k)+elax/(nobs-nullx(k))
         endif
7        continue

c Compute <derz> and <elaz> for each observation in the sample.
c Compute <expderz> and <expelaz>.
         if(nvaz.gt.0)then
            do 8 k=1,nvaz
            ngroup=numlz(k)
            if(ngroup.eq.0)then
c Case without LAMBDA(Z).
               derz=factorz*delta(k)/2
            else
c Case with LAMBDA(Z).
               derz=(factorz*delta(k)/2)*z(i,k)**(vlz(ngroup)-1)
            endif
            elaz=derz*z(i,k)/expy(i)
            expderz(k)=expderz(k)+derz/nobs
8           expelaz(k)=expelaz(k)+elaz/nobs
         endif
         ymean=ymean+y(i)/nobs
         eymean=eymean+expy(i)/nobs
15       continue

c Case where a heteroskedasticity variable Z also appears as an independent
c variable X.
         if(nvaz.gt.0)then
            do 20 k=1,nvaz
            ix=numxeqz(k)
            if(ix.gt.0)then
               expderx(ix)=expderx(ix)+expderz(k)
               expelax(ix)=expelax(ix)+expelaz(k)
               expderz(k)=expderx(ix)
               expelaz(k)=expelax(ix)
            endif
20          continue
         endif

c Sept.99: Compute the Pearson R2E defined as the square of the coefficient
c          of correlation between the observed Y and the expected value of Y.
c          This is the new definition of R2E replacing the old one.
         sumyey=0.d0
         sumy2=0.d0
         sumey2=0.d0
         do 21 i=1,nobs
         devy=y(i)-ymean
```

```
      devey=expy(i)-eymean
      sumyey=sumyey+devy*devey
      sumy2=sumy2+devy**2
      sumey2=sumey2+devey**2
21    continue
      r2e=(sumyey**2)/(sumy2*sumey2)
      return
      end

         subroutine fcn(npar,grad,fun,par,iflag)
c Compute the function fun = - L(par(1),...,par(npar)) in terms of minimum.
c  L is the log-likelihood function concentrated on BETA and SIGW2. It only
c  depends on:
c  par(1) = 0.5*dlog((1+rho1)/(1-rho1)) <=> rho1 = dtanh(par(1))
c  par(2) = 0.5*dlog((1+rho2)/(1-rho2)) <=> rho2 = dtanh(par(2))
c  par(3) = dlog(pi1/(1-pi1))           <=> pi1  = dexp(par(3))/(1+dexp(par(3)))
c  par(4) = dlog(pi2/(1-pi2))           <=> pi2  = dexp(par(4))/(1+dexp(par(4)))
c  where par(1),...,par(4) are unconstrained parameters used in DFP minimization
c  routine which calls this routine:
c    -infinity < par(1) < +infinity
c    -infinity < par(2) < +infinity
c    -infinity < par(3) < +infinity
c    -infinity < par(4) < +infinity
c  and rho1, rho2, pi1 and pi2 are constrained autocorrelation parameters used
c  in this routine:
c    -1 < rho1 < +1
c    -1 < rho2 < +1
c     0 < pi1  < +1
c     0 < pi2  < +1.
c  The next two groups of par(.) are associated with the DELTA-parameters and
c  the LAMBDA(Z)-parameters in the heteroskedasticity function f(Z).
c  The last two groups of par(.) are associated with LAMBDA(Y)-parameter and
c  the LAMBDA(X)-parameters for the dependent (Y) and independent (X) variables
c  respectively in the regression equation.
c NOTE:
c    RHO1 and RHO2 are always estimated, whereas the remaining parameters PI1,
c    PI2 , DELTA's, LAMBDA(Z)'s, LAMBDA(Y) and the LAMBDA(X)'s can be fixed or
c    estimated.
c*******************************************************************************
      implicit double precision (a-h,o-z)
c Parameters used in DIMENSION statements:
c n      - maximum number of observations.
c nyxz    - maximum number of independent variables X (excluding the constant)
c          and heteroskedasticity variables Z in a DATABASE.
c nx     - maximum number of independent variables X (including the constant),
c          used in ESTIMATION.
c nlx    - maximum number of distinct LAMBDA(X)'s (fixed and estimated).
c nz     - maximum number of heteroskedasticity variables Z used in ESTIMATION.
c nlz    - maximum number of distinct LAMBDA(Z)'s (fixed and estimated).
c nxx    - maximum number of different elements in the (nx x nx) symmetric
c          matrix XPPX stored columnwise as an upper triangular matrix.
c nord   - maximum number of autocorrelation orders.
c np     - maximum number of RHO's, PI's, DELTA's, LAMBDA(Z)'s, LAMBDA(Y)
c          and LAMBDA(X)'s in the concentrated log-likelihood function.
c npall  - maximum number of estimated BETA's, SIGW2, RHO's, PI's, DELTA's,
c          LAMBDA(Z)'s, LAMBDA(Y) and LAMBDA(X)'s in the system.
c ncop   - maximum number of different elements in the (npall x npall) symmetric
```

```
c          matrix COV stored columnwise as an upper triangular matrix.
c          COV is the covariance matrix of all the estimated BETA's, SIGW2,
c          RHO's, PI's, DELTA's, LAMBDA(Z)'s, LAMBDA(Y) and LAMBDA(X)'s.
c npcon  - maximum number of estimated BETA's, SIGW2, RHO's, PI's and DELTA's.
c ncond  - maximum number of different elements in the (npcon x npcon) symmetric
c          matrix CONDCOV stored columnwise as an upper triangular matrix.
c          CONDCOV is the covariance matrix of the estimated BETA's, SIGW2,
c          RHO's, PI's and DELTA's, CONDITIONAL on the values (fixed or
c          estimated) of LAMBDA(Y) and LAMBDA(X)'s.
      parameter (n=1000,nyxz=500,nx=50,nlx=10,nz=10,nlz=10,
     +nxx=(nx*(nx+1))/2,nord=2,np=2*nord+nz+nlz+1+nlx, npall=nx+1+np,
     +ncop=(npall*(npall+1))/2, npcon=npall-nlz-1-nlx,
     +ncond=(npcon*(npcon+1))/2)

      dimension grad(*),par(*)
c Variables Y, X and Z:
      dimension y(n),x(n,nx),z(n,nz),xz(nyxz),xbeta(n),ysam(n)
      dimension ybox(n),xbox(n,nx),zbox(n,nz)
      dimension ybfz(n),xbfz(n,nx)
      dimension py(n),px(n,nx),pz(n,nz+1),xppy(nx),xppx(nxx)
      dimension numx(nx),numlx(nx),xmean(nx),xmin(nx),xmax(nx),scalx(nx)
      dimension numz(nz),numlz(nz),zmean(nz),zmin(nz),zmax(nz),scalz(nz)
      dimension xtmean(nx),ztmean(nz+1),xtconst(n)
      dimension coryx(nx,2),coryz(nz,2)
      dimension nullx(nx),nullz(nz),numxeqz(nz)
      dimension smpderx(nx),smpelax(nx),smpderz(nz),smpelaz(nz)
      dimension expy(n),expderx(nx),expelax(nx),expderz(nz),expelaz(nz)
c Residuals U, V and W:
      dimension u(n),sqrtfz(n),v(n),w(n),wp(n)
      dimension derv(n),derw(n),dwmat(n,np)
c Parameter vectors BETA, RHO, PI, DELTA, LAMBDA(Z) and LAMBDA(X).
c Note: SIGW2 and LAMBDA(Y) are scalars so they don't appear in DIMENSION.
      dimension beta(nx),gradbeta(nx)
      dimension rho(nord),gradrho(nord),tracerho(nord)
      dimension pi(nord),gradpi(nord),ntpi(nord),tracepi(nord)
      dimension delta(nz),graddelta(nz),ntdel(nz)
      dimension vlz(nlz),gradlamz(nlz),ntlz(nlz)
      dimension vlx(nlx),gradlamx(nlx),ntlx(nlx)
c Different matrices for autocorrelation:
      dimension r1(n,n),r2(n,n)
      dimension rstar1(n,n),binv1(n,n)
      dimension rstar2(n,n),binv2(n,n)
      dimension p(n,n),pinv(n,n),lw(n),mw(n)
      dimension oldinv(n,n),diffrst(n,n),derstar(n,n)
c Covariance matrices of the estimated parameters:
      dimension derlt(npall),cov(ncop),condcov(ncond)
c Variables of character type:
      character*80 comment,yxzfmt,resfile,matfile(2),yxzfile
      character*30 TabFile
      character*30 ElastFile
      character*10 namy,namx(nx),namz(nz),namxz(nyxz)
      character*9 varname
      character*8 matname(2)
      character*3 status
      character*3 varnum
      character*2 ntypelx
      equivalence (oldinv,diffrst)
```

```
c Parameters declared in NAMELIST/MODEL/ and used in Main Program:
c   nlist   = 0 : No list.
c             1 : List the iterations during the maximization of the
c                 log-likelihood function.
c   nstep   = Maximum number of steps or iterations for the maximization
c             of the log-likelihood function. This number should be set
c             at least equal to the square of the number of parameters
c             being estimated in the model.
c   epsi    = Convergence parameter for the log-likelihood function
c             (Usual value = 1.d-7).
c   wght    = Weighting factor for the relative change of an estimated
c             parameter (Usual value = 1).

c Parameters declared in NAMELIST/MODEL/ and used in this Subroutine (FCN):
c   nobs    = Number of observations used in ESTIMATION.
c   nvaxz   = Number of independent variables X and heteroskedasticity
c             variables Z in the DATABASE used in ESTIMATION.
c   nvax    = Number of independent variables X used in ESTIMATION,
c             excluding the constant.
c   norder  = Number of autocorrelation orders used in ESTIMATION (max. 2).
c           = 0 : No autocorrelation order considered.
c           = 1 : One autocorrelation order.
c           = 2 : Two autocorrelation orders.
c   nvaz    = Number of heteroskedasticity variables Z used in ESTIMATION.
c   nlamz   = 0 : No Box-Cox transformation on the Z's.
c           = Number of distinct LAMBDA(Z)'s (fixed and estimated) used
c             in ESTIMATION.
c   nlamy   = 0 : No Box-Cox transformation on Y.
c           = 1 : One Box-Cox transformation on Y.
c   nlamx   = 0 : No Box-Cox transformation on the X's.
c           = Number of distinct LAMBDA(X)'s (fixed and estimated) used
c             in ESTIMATION.
c   nscale  = 0 : No scaling on Y, X's and Z's.
c           = 1 : Scaling on Y, X's and Z's.
c   ncor    = 0 : No Table of Variance-Decomposition Proportions for the
c                 Independent Variables X (including the Constant) and
c                 No Correlation Matrix for the Independent Variables X
c                 (exluding the Constant) and the Dependent Variable Y.
c           = 1 : Table of Variance-Decomposition Proportions for the
c                 Transformed Independent Variables X (including the Constant)
c                 and Correlation Matrix for the Transformed Independent
c                 Variables X (excluding the Constant) and the Transformed
c                 Dependent Variable Y.
c           = 2 : Same as 1 but for both - Original and Transformed -
c                 Dependent and Independent Variables.

c CONSTANTS declared in DATA:
c   epsinv  = Input constant for SINV used as a relative tolerance for
c             test on loss of significance during inversion of a symmetric
c             positive definite matrix.
c   cstpi   = Mathematical constant PI = 3.14159... (not to be confused with
c             the PI-parameters of the autocorrelation process).
c   rmaxlam = Maximum absolute value of an estimated LAMBDA(Y,X,Z)-parameter
c             used as the lower and upper bounds: -rmaxlam <= LAMBDA <= rmaxlam.
      real*4 epsi,wght
      common nlist,nstep,epsi,wght,noprint
      namelist/model/nlist,nstep,epsi,wght,nobs,namy,nvaxz,nvax,norder,
```

```
     +nvaz,nlamz,nlamy,nlamx,nscale,nexpy,nres,ncor,ncov,varname,
     +nvariant
      namelist/autocor/rho,pi,ntpi,matfile
      namelist/hetero/numz,numlz,delta,ntdel,vlz,ntlz
      namelist/yboxcox/vly,ntly,numlxy
      namelist/xboxcox/numx,numlx,vlx,ntlx
      namelist/yxzdata/yxzfile,yxzfmt
      data epsinv/1.d-7/
      data cstpi/3.1415926535898d0/
      data rmaxlam/10.d0/

      if(iflag.eq.1)then
c***********************************************************************
c     BEGINNING OF IFLAG = 1
c***********************************************************************
         read model
c Create the name of a .tab file containing the estimated results of a variant
c which will be used as input to TABLEX program
      ncar=LNBLNK(varname)
      if(nvariant.lt.10)then
            encode(16,101,varnum)nvariant
101         format(i1)
            TabFile=varname(1:ncar)//varnum(1:1)//'.tab'
            ElastFile=varname(1:ncar)//varnum(1:1)//'_ELA'//'.txt'
      elseif(nvariant.lt.100)then
            encode(16,102,varnum)nvariant
102         format(i2)
            TabFile=varname(1:ncar)//varnum(1:2)//'.tab'
            ElastFile=varname(1:ncar)//varnum(1:2)//'_ELA'//'.txt'
      else
            encode(16,103,varnum)nvariant
103         format(i3)
            TabFile=varname(1:ncar)//varnum(1:3)//'.tab'
            ElastFile=varname(1:ncar)//varnum(1:3)//'_ELA'//'.txt'
      endif

        if(noprint.eq.0)print model

         if(norder.gt.0)then
c Read the initial values of RHO's and PI's and the type NTPI of PI's:
c NTPI = 0 if PI is fixed, or 1 if PI is estimated.
            read autocor
            if(noprint.eq.0)print autocor
c Read the first impact matrix R1 from 'matfile(1)' and
c compute the sum of each row in R1.
c 'matfile(1)' contains on the first line:
c 'matname(1)'= the name of the matrix R1.
            open(unit=1,file=matfile(1))
              read(1,1001)matname(1)
            do 3 i=1,nobs
               read(1,1010)(r1(i,j),j=1,nobs)
               rowsum=0.d0
               do 1 j=1,nobs
    1          rowsum=rowsum+r1(i,j)
c Row normalize the matrix R1.
c The original matrix R1 is destroyed and replaced by the row normalized matrix.
c FEB 22 1994: Special case: if all the elements in a row are zero, i.e. rowsum
```

```
c = 0, then they are left as such, without being divided by rowsum.
               if(rowsum.gt.0.d0)then
                  do 2 j=1,nobs
    2             r1(i,j)=r1(i,j)/rowsum
               endif
    3       continue
            close(1)
c Convert R1 from double to single dimension array so that the matrix can be
c used in the subroutines which work only with single dimension arrays.
            call array(2,nobs,nobs,n,n,r1,r1)

c Read the second impact matrix R2 from 'matfile(2)' and
c compute the sum of each row in R2.
c 'matfile(2)' contains on the first line:
c 'matname(2) = the name of the matrix R2.
            if(norder.eq.2)then
               open(unit=1,file=matfile(2))
                 read(1,1001)matname(2)
               do 6 i=1,nobs
                  read(1,1010)(r2(i,j),j=1,nobs)
                  rowsum=0.d0
                  do 4 j=1,nobs
    4             rowsum=rowsum+r2(i,j)
c Row normalize the matrix R2.
c The original matrix R2 is destroyed and replaced by the row normalized matrix.
c FEB 22 1994: Special case: if all the elements in a row are zero, i.e. rowsum
c = 0, then they are left as such, without being divided by rowsum.
                  if(rowsum.gt.0.d0)then
                     do 5 j=1,nobs
    5                r2(i,j)=r2(i,j)/rowsum
                  endif
    6          continue
               close(1)
c Convert R2 from double to single dimension array so that the matrix can be
c used in the subroutines which work only with single dimension arrays.
               call array(2,nobs,nobs,n,n,r2,r2)
            endif
         else
c Initialize P=I (Identity matrix) for the case without autocorrelation.
            do 7 i=1,nobs
               do 7 j=1,nobs
                  p(i,j)=0.d0
                  if(i.eq.j)p(i,j)=1.d0
    7       continue
c Convert P from double to single dimension array so that the matrix can be
c used in the subroutines which work only with single dimension arrays.
            call array(2,nobs,nobs,n,n,p,p)
         endif

c Read the indices NUMZ of the heteroskedasticity variables Z's used for
c estimation.
c Read the indices NUMLZ of the LAMBDA(Z)'s associated with the variables Z's:
c    numlz = 0 if Z is not transformed by Box-Cox.
c          = Positive integer representing the index of the LAMBDA(Z).
c Read the type NTDEL and initial value DELTA of DELTA:
c    ntdel = 0 if DELTA is fixed, or 1 if DELTA is estimated.
c    delta = Fixed value if ntdel = 0, or Initial value if ntdel = 1.
```

```
c Read the type NTLZ and initial value VLZ of LAMBDA(Z):
c    ntlz  = 0 if LAMBDA(Z) is fixed, or 1 if LAMBDA(Z) is estimated.
c    vlz   = Fixed value if ntlz = 0, or Initial value if ntlz = 1.
c Compute NESTDELTA : Total number of estimated DELTA's.
c Compute NFIXDELTA : Total number of fixed DELTA's.
c Compute NESTLAMZ : Total number of estimated LAMBDA(Z)'s.
c Compute NFIXLAMZ : Total number of fixed LAMBDA(Z)'s.
         nestdelta=0
         nfixdelta=0
         nestlamz=0
         nfixlamz=0
         if(nvaz.gt.0)then
            read hetero
            if(noprint.eq.0)print hetero
            do 8 i=1,nvaz
    8       if(ntdel(i).eq.1)nestdelta=nestdelta+1
            if(nlamz.gt.0)then
               do 9 i=1,nlamz
    9          if(ntlz(i).eq.1)nestlamz=nestlamz+1
            endif
            nfixdelta=nvaz-nestdelta
            nfixlamz=nlamz-nestlamz
         endif

c Read the type NTLY and initial value VLY of LAMBDA(Y).
c  ntly   = 0 if LAMBDA(Y) is fixed, or 1 if LAMBDA(Y) is estimated.
c  vly    = Fixed value if ntly =0, or Initial value if ntly = 1.
c Read the index NUMLXY of the LAMBDA(X) which is set equal to LAMBDA(Y).
c  numlxy = 0 if there is no LAMBDA(X) set equal to LAMBDA(Y).
c         = Positive index of the LAMBDA(X) set equal to LAMBDA(Y).
         ntly=0
         numlxy=0
         if(nlamy.eq.1)then
            read yboxcox
            if(noprint.eq.0)print yboxcox
         endif

c Read the indices NUMX of the independent variables X's used for
c estimation.
c Read the indices NUMLX of the LAMBDA(X)'s associated with the variables X's:
c    numlx = 0 if X is not transformed by Box-Cox.
c          = Positive integer representing the index of the LAMBDA(X).
c Read the type NTLX and initial value VLX of LAMBDA(X).
c    ntlx  = 0 if LAMBDA(X) is fixed, or 1 if LAMBDA(X) is estimated.
c    vlx   = Fixed value if ntlx = 0, or Initial value if ntlx = 1.
c Compute NESTLAMX : Total number of estimated LAMBDA(X)'s.
c Compute NFIXLAMX : Total number of fixed LAMBDA(X)'s.
         nb=nvax+1
         nestlamx=0
         nfixlamx=0
         read xboxcox
c Correction 17 nov 2012
         if(noprint.eq.0)print xboxcox
         if(nlamx.gt.0)then
            do 10 i=1,nlamx
   10       if(ntlx(i).eq.1)nestlamx=nestlamx+1
            nfixlamx=nlamx-nestlamx
```

```
         endif

c Read the dependent, independent variables (Y,X) and heteroskedasticity
c variables Z, excluding the constant, from 'yxzfile'. Note that the dependent
c variable Y should appear in the first column, and the variables X and Z in
c the following columns. The variables X and Z can appear in any order.
c The constant is generated in the program as the last independent variable
c in the group of variables X used in ESTIMATION.
c Compute the minimum, maximum and mean values of Y, X and Z used in
c ESTIMATION, in original units.
         read yxzdata
         if(noprint.eq.0)print yxzdata
         open(unit=1,file=yxzfile)
         read(1,1001)namy,(namxz(k),k=1,nvaxz)
         ymean=0.d0
         ymin=1.d300
         ymax=-ymin
         nully=0
         do 11 k=1,nvax
            xmean(k)=0.d0
            xmin(k)=1.d300
            xmax(k)=-xmin(k)
   11    nullx(k)=0
         namx(nb)='CONSTANT'
         xmean(nb)=1.d0
         xmin(nb)=1.d0
         xmax(nb)=1.0
         nullx(nb)=0
         if(nvaz.gt.0)then
            do 12 k=1,nvaz
               zmean(k)=0.d0
               zmin(k)=1.d300
               zmax(k)=-zmin(k)
   12       nullz(k)=0
         endif
         do 15 i=1,nobs
            read(1,yxzfmt)y(i),(xz(k),k=1,nvaxz)
            ymean=ymean+y(i)/nobs
            if(y(i).lt.ymin)ymin=y(i)
            if(y(i).gt.ymax)ymax=y(i)
            if(y(i).eq.0.d0)nully=nully+1
            x(i,nb)=1.d0
            do 13 k=1,nvax
               ik=numx(k)
               x(i,k)=xz(ik)
               namx(k)=namxz(ik)
               xmean(k)=xmean(k)+x(i,k)/nobs
               if(x(i,k).lt.xmin(k))xmin(k)=x(i,k)
               if(x(i,k).gt.xmax(k))xmax(k)=x(i,k)
   13       if(x(i,k).eq.0.d0)nullx(k)=nullx(k)+1
            if(nvaz.gt.0)then
               do 14 k=1,nvaz
                  ik=numz(k)
                  z(i,k)=xz(ik)
                  namz(k)=namxz(ik)
                  zmean(k)=zmean(k)+z(i,k)/nobs
                  if(z(i,k).lt.zmin(k))zmin(k)=z(i,k)
```

```
                  if(z(i,k).gt.zmax(k))zmax(k)=z(i,k)
   14          if(z(i,k).eq.0.d0)nullz(k)=nullz(k)+1
            endif
   15    continue
         close(1)

      open(unit=4,file=ElastFile)
      write(4,1404)varname,nvariant,namy,nobs
1404  format(a9,i4/a10,i5)

c APRIL 19 2001
c Compute the Table of Variance-Decomposition Proportions for the Original
c         Independent Variables (including the Constant) and
c         the Correlation Matrix for the Original Independent Variables X
c         (excluding the Constant) and the Dependent Variable Y.
         if(ncor.eq.2)call vardec(x,xppx,namx,n,nobs,nb,nxx)
         do 16 i=1,nobs
   16    x(i,nb)=y(i)
         xmean(nb)=ymean
         call correl(x,xmean,xppy,xppx,namx,namy,n,nobs,nb,nxx,ncor)
         do 17 i=1,nobs
   17    x(i,nb)=1.0d0
         xmean(nb)=1.0d0
         nbb=(nb*(nb-1))/2
         do 104 i=1,nb-1
  104    coryx(i,1)=xppx(nbb+i)
         coryx(nb,1)=0.d0
c JULY 2003
c Compute the correlations between Original Y and Z_m (m=1,nvaz): coryz
      if(nvaz.gt.0)then
         nvazy=nvaz+1
         do 130 i=1,nobs
  130    z(i,nvazy)=y(i)
         zmean(nvazy)=ymean
         call correl(z,zmean,xppy,xppx,namz,namy,n,nobs,nvazy,nzz,ncor)
         nbb=(nvazy*(nvazy-1))/2
         do 131 i=1,nvaz
  131    coryz(i,1)=xppx(nbb+i)
      endif
c Compute the scaling factors for Y and X: SCALY and SCALX.
c Print the minimum, mean and maximum values, the number of null observations
c and the scaling factors for Y and X

         vmax=dmax1(dabs(ymin),dabs(ymax))
         scaly=1.d0/10.d0**idint(dlog10(vmax))
      if(noprint.eq.0)then
         print 1030
         print 1040,namy,ymin,ymean,ymax,nully,scaly
         print 1050
      endif
         do 18 k=1,nb
            vmax=dmax1(dabs(xmin(k)),dabs(xmax(k)))
            scalx(k)=1.d0/10.d0**idint(dlog10(vmax))
            if(noprint.eq.0)print 1060,namx(k),xmin(k),xmean(k),xmax(k)
     +      ,nullx(k),scalx(k)
   18    continue
```

```
c If NSCALE=1: Scale the dependent, independent and heteroskedasticity variables
c Y, X and Z.
c For IFLAG=1,2 or 4, all computations are made with scaled variables if
c NSCALE=1.
c Initialize YBOX and XBOX with Y and X.
c Compute SUMLNY = sum of ln(Y) if there is a Box-Cox transformation on Y.
         sumlny=0.d0
         do 19 i=1,nobs
            if(nscale.eq.1)y(i)=scaly*y(i)
            ybox(i)=y(i)
            if(nlamy.eq.1)sumlny=sumlny+dlog(y(i))
            do 19 k=1,nb
               if(nscale.eq.1)x(i,k)=scalx(k)*x(i,k)
               xbox(i,k)=x(i,k)
   19    continue

c Compute the fixed Box-Cox transformation on Y only once since they don't
c change during the iterations.
         if(nlamy.eq.1.and.ntly.eq.0)then
            do 20 i=1,nobs
   20       ybox(i)=zlam(y(i),vly)
         endif

c Compute the fixed Box-Cox transformations on X's only once since they don't
c change during the iterations.
         if(nfixlamx.gt.0)then
            do 22 k=1,nvax
               if(numlx(k).gt.0)then
                  ngroup=numlx(k)
                  if(ntlx(ngroup).eq.0)then
                     do 21 i=1,nobs
   21                xbox(i,k)=zlam(x(i,k),vlx(ngroup))
                  endif
               endif
   22       continue
         endif

c Compute the scaling factors for Z: SCALZ.
c Print the minimum, mean and maximum values, the number of null observations
c and the scaling factors for Z.
         if(nvaz.gt.0)then
            if(noprint.eq.0)print 1070
            do 24 k=1,nvaz
c Compute NUMXEQZ: entry index of an independent variable X which appears also
c as a heteroskedasticity variable Z. The index corresponds to the order in
c which X is entered among the NVAX independent variables.
               numxeqz(k)=0
               do 23 j=1,nvax
   23          if(numx(j).eq.numz(k))numxeqz(k)=j
               vmax=dmax1(dabs(zmin(k)),dabs(zmax(k)))
               scalz(k)=1.d0/10.d0**idint(dlog10(vmax))
               if(noprint.eq.0)print 1060,namz(k),zmin(k),zmean(k),
     +         zmax(k),nullz(k),scalz(k)
   24       continue
c Initialize ZBOX with Z.
            do 25 i=1,nobs
               do 25 k=1,nvaz
```

```
                  if(nscale.eq.1)z(i,k)=scalz(k)*z(i,k)
   25       zbox(i,k)=z(i,k)
c Compute the fixed Box-Cox transformations on Z's only once since they don't
c change during the iterations.
            if(nfixlamz.gt.0)then
               do 27 k=1,nvaz
                  if(numlz(k).gt.0)then
                     ngroup=numlz(k)
                     if(ntlz(ngroup).eq.0)then
                        do 26 i=1,nobs
   26                   zbox(i,k)=zlam(z(i,k),vlz(ngroup))
                     endif
                  endif
   27          continue
            endif
         endif

c Compute the total number NPAR of estimated RHO's, PI's, DELTA's,
c LAMBDA(Z)'s, LAMBDA(Y) and LAMBDA(X)'s.
c Initialization of vectors PAR and GRAD.
c For estimated RHO's and PI's.
         kpar=0
         if(norder.gt.0)then
            do 28 i=1,norder
               kpar=kpar+1
   28       par(kpar)=0.5*dlog((1+rho(i))/(1-rho(i)))
            do 29 i=1,norder
               if(ntpi(i).eq.1)then
                  kpar=kpar+1
c Due to the change of parameter for each estimated PI:
c  0 < PI < 1  <=>  -infinity < PAR < +infinity,
c the initial value of PI cannot be exactly equal to 0 or 1,
c if PI is to be estimated.
c If the initial value of PI is 0, this value is replaced by 0.01.
c If the initial value of PI is 1, this value is replaced by 0.99.
                  if(pi(i).eq.0.d0)pi(i)=0.01d0
                  if(pi(i).eq.1.d0)pi(i)=0.99d0
                  par(kpar)=dlog(pi(i)/(1-pi(i)))
               endif
   29       continue
         endif

c For estimated DELTA's and LAMBDA(Z)'s.
         if(nestdelta.gt.0)then
            do 30 i=1,nvaz
               if(ntdel(i).eq.1)then
                  kpar=kpar+1
                  par(kpar)=delta(i)
               endif
   30       continue
         endif
         if(nestlamz.gt.0)then
            do 31 i=1,nlamz
               if(ntlz(i).eq.1)then
                  kpar=kpar+1
                  par(kpar)=vlz(i)
               endif
```

```
   31       continue
         endif

c For estimated LAMBDA(Y) if it is estimated distinctly from the LAMBDA(X)'s.
         if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.0)then
            kpar=kpar+1
            par(kpar)=vly
         endif
c For estimated LAMBDA(X)'s.
         if(nestlamx.gt.0)then
            do 32 i=1,nlamx
               if(ntlx(i).eq.1)then
                  kpar=kpar+1
                  par(kpar)=vlx(i)
               endif
   32       continue
         endif
c Initialize NPAR = Total number of estimated RHO, PI, DELTA, LAMBDA(Z),
c LAMBDA(Y) and LAMBDA(X).
         npar=kpar
         do 33 i=1,npar
   33    grad(i)=0.d0
c Compute the constant of the concentrated Log-likelihood function FUN.
         conlik=nobs*(1+dlog(2*cstpi))/2
      endif
c**********************************************************************
c     END OF IFLAG = 1
c     BEGINNING OF IFLAG = 2, 3 AND 4
c**********************************************************************
c Update RHO and PI with current values in vector PAR for estimated
c RHO's and PI's.
      kpar=0
      if(norder.gt.0)then
         do 34 i=1,norder
            kpar=kpar+1
            rho(i)=dtanh(par(kpar))
c Test if each estimated RHO is equal to 1.
            if(dabs(rho(i)).eq.1.d0)then
               fun=2*fun
               return
            endif
   34    continue
         do 35 i=1,norder
            if(ntpi(i).eq.1)then
               kpar=kpar+1
               expar=dexp(par(kpar))
               pi(i)=expar/(1+expar)
c Test if each estimated PI is approaching zero.
               if(pi(i).lt.1.d-10)then
                  fun=2*fun
                  return
               endif
            endif
   35    continue

c Compute RSTAR1 = PI1 * INVERSE(B1) where B1 = [ INVERSE(R1) - (1- PI1)*I ].
         if(iflag.eq.1.or.(iflag.ne.1.and.ntpi(1).eq.1))then
```

```
            c=pi(1) - 1.
            call shminv(r1,c,oldinv,binv1,nobs)
            call smpy(binv1,pi(1),rstar1,nobs,nobs,0)
         endif

c Compute RSTAR2 = PI2 * INVERSE(B2) where B2 = [ INVERSE(R2) - (1- PI2)*I ].
         if(norder.eq.2)then
            if(iflag.eq.1.or.(iflag.ne.1.and.ntpi(2).eq.1))then
               c=pi(2) - 1.
               call shminv(r2,c,oldinv,binv2,nobs)
               call smpy(binv2,pi(2),rstar2,nobs,nobs,0)
            endif
         endif

c Compute matrix P = [ I - RHO(1)*RSTAR1 ] if norder=1,
c or P = [ I - RHO(1)*RSTAR1 - RHO(2)*RSTAR2 ] if norder=2.
         call pmatrix(rho,rstar1,rstar2,p,nobs,norder)

c Compute the inverse PINV and determinant DETP of P using subroutine MINV
c for a GENERAL MATRIX (non-symmetric or symmetric).
         call mcpy(p,pinv,nobs,nobs,0)
         call minv(pinv,nobs,detp,lw,mw,ier)
c Test ier not equal to 0 (matrix pinv not invertible).
         if(ier.ne.0)then
            fun=2*fun
            return
         endif
      endif

c Update DELTA and VLZ with current values in vector PAR for estimated
c DELTA's and LAMBDA(Z)'s.
      if(nestdelta.gt.0)then
         do 36 i=1,nvaz
            if(ntdel(i).eq.1)then
               kpar=kpar+1
               delta(i)=par(kpar)
            endif
   36    continue
      endif
      if(nestlamz.gt.0)then
         do 37 i=1,nlamz
            if(ntlz(i).eq.1)then
               kpar=kpar+1
               vlz(i)=par(kpar)
               if(dabs(vlz(i)).gt.rmaxlam)then
                  fun=2*fun
                  return
               endif
            endif
   37    continue
      endif

c Update VLY and VLX with current values in vector PAR for estimated
c LAMBDA(Y) and LAMBDA(X)'s.
      if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.0)then
         kpar=kpar+1
         vly=par(kpar)
```

```
         if(dabs(vly).gt.rmaxlam)then
            fun=2*fun
            return
         endif
      endif
      if(nestlamx.gt.0)then
         do 38 i=1,nlamx
            if(ntlx(i).eq.1)then
               kpar=kpar+1
               vlx(i)=par(kpar)
               if(dabs(vlx(i)).gt.rmaxlam)then
                  fun=2*fun
                  return
               endif
               if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.i)vly=vlx(i)
            endif
   38    continue
      endif

c For IFLAG=3 only:
c    Convert the variables Y, X and Z from scaled to original units.
c    All computations are made with the variables in original units.
c    Initialize YBOX and XBOX with original Y and X.
c    Recompute SUMLNY = sum of ln(Y) in original units.
      if(iflag.eq.3)then
         sumlny=0.d0
         do 39 i=1,nobs
            if(nscale.eq.1)y(i)=y(i)/scaly
            ybox(i)=y(i)
            if(nlamy.eq.1)sumlny=sumlny+dlog(y(i))
            do 39 k=1,nb
               if(nscale.eq.1)x(i,k)=x(i,k)/scalx(k)
               xbox(i,k)=x(i,k)
   39    continue
c Compute the fixed Box-Cox transformation on original Y.
         if(nlamy.eq.1.and.ntly.eq.0)then
            do 40 i=1,nobs
   40       ybox(i)=zlam(y(i),vly)
         endif
c Compute the fixed Box-Cox transformations on original X's.
         if(nfixlamx.gt.0)then
            do 42 k=1,nvax
               if(numlx(k).gt.0)then
                  ngroup=numlx(k)
                  if(ntlx(ngroup).eq.0)then
                     do 41 i=1,nobs
   41                xbox(i,k)=zlam(x(i,k),vlx(ngroup))
                  endif
               endif
   42       continue
         endif
C Initialize ZBOX with original Z.
         if(nvaz.gt.0)then
            do 43 i=1,nobs
               do 43 k=1,nvaz
                  if(nscale.eq.1)z(i,k)=z(i,k)/scalz(k)
                  zbox(i,k)=z(i,k)
```

```
   43       continue
c Compute the fixed Box-Cox transformations on original Z's.
            if(nfixlamz.gt.0)then
               do 45 k=1,nvaz
                  if(numlz(k).gt.0)then
                     ngroup=numlz(k)
                     if(ntlz(ngroup).eq.0)then
                        do 44 i=1,nobs
   44                   zbox(i,k)=zlam(z(i,k),vlz(ngroup))
                     endif
                  endif
   45          continue
            endif
c Convert all the DELTA's (fixed and estimated) from scaled to original units
c of Z.
            if(nscale.eq.1)then
               do 46 k=1,nvaz
                  if(numlz(k).gt.0)then
                     ngroup=numlz(k)
                     delta(k)=delta(k)*scalz(k)**vlz(ngroup)
                  else
                     delta(k)=delta(k)*scalz(k)
                  endif
   46          continue
            endif
         endif
      endif

c For IFLAG=1,2,3 or 4, compute the estimated Box-Cox transformations on Y
c and X at each iteration.
      if(nlamy.eq.1.and.ntly.eq.1)then
         do 47 i=1,nobs
   47    ybox(i)=zlam(y(i),vly)
      endif
      if(nestlamx.gt.0)then
         do 49 k=1,nvax
            if(numlx(k).gt.0)then
               ngroup=numlx(k)
               if(ntlx(ngroup).eq.1)then
                  do 48 i=1,nobs
   48             xbox(i,k)=zlam(x(i,k),vlx(ngroup))
               endif
            endif
   49    continue
      endif

c For IFLAG=1,2,3 or 4, compute the estimated Box-Cox transformations on Z's
c at each iteration.
      if(nvaz.gt.0)then
c Case with HETEROSKEDASTICITY.
         if(nestlamz.gt.0)then
            do 51 k=1,nvaz
               if(numlz(k).gt.0)then
                  ngroup=numlz(k)
                  if(ntlz(ngroup).eq.1)then
                     do 50 i=1,nobs
   50                zbox(i,k)=zlam(z(i,k),vlz(ngroup))
```

```
                  endif
               endif
   51       continue
         endif
c Compute SQRTFZ: square root of the HETEROSKEDASTICITY function f(Z).
c Compute SUMLNFZ: sum of ln f(Z) = sum of DELTA*ZBOX.
         sumlnfz=0.d0
         do 53 i=1,nobs
         sum=0.d0
         do 52 k=1,nvaz
   52    sum=sum+delta(k)*zbox(i,k)
         sqrtfz(i)=dexp(sum/2.)
   53    sumlnfz=sumlnfz+sum
c Divide YBOX and XBOX by SQRTFZ and store the transformed variables in
c YBFZ and XBFZ.
         do 54 i=1,nobs
         ybfz(i)=ybox(i)/sqrtfz(i)
         do 54 k=1,nb
   54    xbfz(i,k)=xbox(i,k)/sqrtfz(i)
      else
c Case without HETEROSKEDASTICITY.
         do 55 i=1,nobs
         sqrtfz(i)=1.d0
         ybfz(i)=ybox(i)
         do 55 k=1,nb
   55    xbfz(i,k)=xbox(i,k)
      endif
c Convert XBFZ from double to single dimension array so that the matrix can
c be used in the subroutines which work only with single dimension arrays.
      call array(2,nobs,nb,n,nx,xbfz,xbfz)

c Compute PY = P*YBFZ and PX = P*XBFZ.
      if(norder.gt.0)then
         call gmprd(p,ybfz,py,nobs,nobs,1)
         call gmprd(p,xbfz,px,nobs,nobs,nb)
      else
         call mcpy(ybfz,py,nobs,1,0)
         call mcpy(xbfz,px,nobs,nb,0)
      endif
c Compute X'P'PY, X'P'PX and BETA = (INVERSE(X'P'PX))*(X'P'PY).
      call gtprd(px,py,xppy,nobs,nb,1)
      call mata(px,xppx,nobs,nb,0)
      call sinv(xppx,nb,epsinv,ier)
      call mprd(xppx,xppy,beta,nb,nb,1,0,1)
c Compute residuals V = YBFZ - XBFZ*BETA and W = P*V.
      call gmprd(xbfz,beta,xbeta,nobs,nb,1)
      call gmsub(ybfz,xbeta,v,nobs,1)
      if(norder.gt.0)then
         call gmprd(p,v,w,nobs,nobs,1)
      else
         call mcpy(v,w,nobs,1,0)
      endif
c September 29 2008
c Compute Mean of Residual W: wmean
      wmean=0.d0
      do 155 i=1,nobs
  155 wmean=wmean+w(i)/nobs
```

```
c Compute residuals U = SQRTFZ*V.
      if(iflag.eq.1.or.iflag.eq.3)then
         do 56 i=1,nobs
   56    u(i)=sqrtfz(i)*v(i)
      endif
c Compute SIGW2 = W'W / NOBS.
      call mata(w,sigw2,nobs,1,0)
      sigw2=sigw2/nobs

c Compute FUN = - L in terms of minimum.
      fun=conlik+nobs*dlog(sigw2)/2
      if(norder.gt.0)fun=fun-dlog(dabs(detp))
      if(nvaz.gt.0)fun=fun+sumlnfz/2
      if(nlamy.eq.1)fun=fun-(vly-1)*sumlny

c If iflag = 1, 2 or 3, compute the first derivatives of FUN with respect to
c the unconstrained parameters PAR:
c    grad(1) = - dL / dpar(1) = - (dL / drho1) * (drho1 / dpar(1))
c    grad(2) = - dL / dpar(2) = - (dL / drho2) * (drho2 / dpar(2))
c    grad(3) = - dL / dpar(3) = - (dL / dpi1) * (dpi1 / dpar(3))
c    grad(4) = - dL / dpar(4) = - (dL / dpi2) * (dpi2 / dpar(4))
c where the first derivatives of L with respect to the constrained parameters
c RHO's and PI's are stored in GRADRHO and GRADPI respectively:
c    gradrho(1) = dL / drho1
c    gradrho(2) = dL / drho2
c    gradpi(1) = dL / dpi1
c    gradpi(2) = dL / dpi2.

      if(iflag.ne.4)then
         kpar=0
         if(norder.gt.0)then
c Compute the first derivative of L with respect to RHO1: gradrho(1).
c Compute the first derivative of FUN with respect to PAR(1): grad(1).
            kpar=kpar+1
            tracerho(1)=-trace(rstar1,pinv,nobs)
            call gmprd(rstar1,v,derw,nobs,nobs,1)
            call smpy(derw,-1.d0,derw,nobs,1,0)
            if(iflag.eq.3)call stormat(derw,dwmat,n,nobs,np,kpar)
            call gtprd(w,derw,wderw,nobs,1,1)
            gradrho(1)=tracerho(1)-wderw/sigw2
            grad(kpar)=-gradrho(1)*(1-rho(1)**2)
c Compute the first derivative of L with respect to RHO2: gradrho(2).
c Compute the first derivative of FUN with respect to PAR(2): grad(2).
            if(norder.eq.2)then
               kpar=kpar+1
               tracerho(2)=-trace(rstar2,pinv,nobs)
               call gmprd(rstar2,v,derw,nobs,nobs,1)
               call smpy(derw,-1.d0,derw,nobs,1,0)
               if(iflag.eq.3)call stormat(derw,dwmat,n,nobs,np,kpar)
               call gtprd(w,derw,wderw,nobs,1,1)
               gradrho(2)=tracerho(2)-wderw/sigw2
               grad(kpar)=-gradrho(2)*(1-rho(2)**2)
            endif
c Compute the first derivative of L with respect to PI1: gradpi(1).
c Compute the first derivative of FUN with respect to PAR(3): grad(3).
            if(ntpi(1).eq.1)then
               kpar=kpar+1
```

```
               call diffir(rstar1,diffrst,nobs)
               call gmprd(diffrst,binv1,derstar,nobs,nobs,nobs)
               c=-rho(1)
               tracepi(1)=c*trace(derstar,pinv,nobs)
               call gmprd(derstar,v,derw,nobs,nobs,1)
               call smpy(derw,c,derw,nobs,1,0)
               if(iflag.eq.3)call stormat(derw,dwmat,n,nobs,np,kpar)
               call gtprd(w,derw,wderw,nobs,1,1)
               gradpi(1)=tracepi(1)-wderw/sigw2
               grad(kpar)=-gradpi(1)*pi(1)*(1-pi(1))
            endif
c Compute the first derivative of L with respect to PI2: gradpi(2).
c Compute the first derivative of FUN with respect to PAR(4): grad(4).
            if(norder.eq.2.and.ntpi(2).eq.1)then
               kpar=kpar+1
               call diffir(rstar2,diffrst,nobs)
               call gmprd(diffrst,binv2,derstar,nobs,nobs,nobs)
               c=-rho(2)
               tracepi(2)=c*trace(derstar,pinv,nobs)
               call gmprd(derstar,v,derw,nobs,nobs,1)
               call smpy(derw,c,derw,nobs,1,0)
               if(iflag.eq.3)call stormat(derw,dwmat,n,nobs,np,kpar)
               call gtprd(w,derw,wderw,nobs,1,1)
               gradpi(2)=tracepi(2)-wderw/sigw2
               grad(kpar)=-gradpi(2)*pi(2)*(1-pi(2))
            endif
         endif

c Compute WP = W'P which is a row vector used in the first derivatives of L
c with respect to DELTA, LAMBDA(Z), LAMBDA(Y) and LAMBDA(X).
         if(nestdelta.gt.0.or.nestlamz.gt.0.or. (nlamy.eq.1.and.ntly.eq.
     +   1).or.nestlamx.gt.0)then
            if(norder.gt.0)then
               call gtprd(w,p,wp,nobs,1,nobs)
            else
               call mcpy(w,wp,1,nobs,0)
            endif
         endif
c Compute the first derivative of L with respect to DELTA: graddelta(.).
c Compute the first derivative of FUN with respect to DELTA: grad(.).
         if(nestdelta.gt.0)then
            do 58 k=1,nvaz
               graddelta(k)=0.d0
               if(ntdel(k).eq.1)then
                  kpar=kpar+1
                  sumzbox=0.d0
                  do 57 i=1,nobs
                     sumzbox=sumzbox+zbox(i,k)
   57             derv(i)=-v(i)*zbox(i,k)/2
                  call gmprd(wp,derv,wderw,1,nobs,1)
                  graddelta(k)=-wderw/sigw2-sumzbox/2
                  grad(kpar)=-graddelta(k)
                  if(iflag.eq.3)then
                     if(norder.gt.0)then
                        call gmprd(p,derv,derw,nobs,nobs,1)
                     else
                        call mcpy(derv,derw,nobs,1,0)
```

```
                  endif
                  call stormat(derw,dwmat,n,nobs,np,kpar)
               endif
            endif
   58       continue
         endif
c Compute the first derivative of L with respect to LAMBDA(Z): gradlamz(.).
c Compute the first derivative of FUN with respect to LAMBDA(Z): grad(.).
         if(nestlamz.gt.0)then
            do 62 i=1,nlamz
               gradlamz(i)=0.d0
               if(ntlz(i).eq.1)then
                  kpar=kpar+1
                  do 59 j=1,nobs
   59             derv(j)=0.d0
                  sumddzl=0.d0
                  do 61 k=1,nvaz
                     if(numlz(k).eq.i)then
                        do 60 j=1,nobs
                           ddzl=delta(k)*derzl(z(j,k),vlz(i))
                           sumddzl=sumddzl+ddzl
   60                   derv(j)=derv(j)-v(j)*ddzl/2
                     endif
   61             continue
                  call gmprd(wp,derv,wderw,1,nobs,1)
                  gradlamz(i)=-wderw/sigw2-sumddzl/2
                  grad(kpar)=-gradlamz(i)
                  if(iflag.eq.3)then
                     if(norder.gt.0)then
                        call gmprd(p,derv,derw,nobs,nobs,1)
                     else
                        call mcpy(derv,derw,nobs,1,0)
                     endif
                     call stormat(derw,dwmat,n,nobs,np,kpar)
                  endif
               endif
   62       continue
         endif
c Compute the first derivative of L with respect to LAMBDA(Y): gradlamy.
c Compute the first derivative of FUN with respect to LAMBDA(Y): grad(.).
c Case where LAMBDA(Y) is estimated distinctly from the group of estimated
c LAMBDA(X)'s.
         if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.0)then
            kpar=kpar+1
            do 63 i=1,nobs
   63       derv(i)=derzl(y(i),vly)
c Heteroskedasticity.
            if(nvaz.gt.0)then
               do 64 i=1,nobs
   64          derv(i)=derv(i)/sqrtfz(i)
            endif
            call gmprd(wp,derv,wderw,1,nobs,1)
            gradlamy=-wderw/sigw2 + sumlny
            grad(kpar)=-gradlamy
            if(iflag.eq.3)then
               if(norder.gt.0)then
                  call gmprd(p,derv,derw,nobs,nobs,1)
```

```
               else
                  call mcpy(derv,derw,nobs,1,0)
               endif
               call stormat(derw,dwmat,n,nobs,np,kpar)
            endif
         endif
c Compute the first derivative of L with respect to LAMBDA(X): gradlamx(.).
c Compute the first derivative of FUN with respect to LAMBDA(X): grad(.).
         if(nestlamx.gt.0)then
            do 70 i=1,nlamx
               gradlamx(i)=0.d0
               if(ntlx(i).eq.1)then
                  kpar=kpar+1
                  do 65 j=1,nobs
   65             derv(j)=0.d0
c Case where the estimated LAMBDA(Y) is set equal to one of the estimated
c LAMBDA(X)'s.
                  if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.i)then
                     do 66 j=1,nobs
   66                derv(j)=derzl(y(j),vly)
                  endif
                  do 68 k=1,nvax
                     if(numlx(k).eq.i)then
                        do 67 j=1,nobs
   67                   derv(j)=derv(j)-beta(k)*derzl(x(j,k),vlx(i))
                     endif
   68             continue
c Heteroskedasticity.
                  if(nvaz.gt.0)then
                     do 69 j=1,nobs
   69                derv(j)=derv(j)/sqrtfz(j)
                  endif
                  call gmprd(wp,derv,wderw,1,nobs,1)
                  gradlamx(i)=-wderw/sigw2
                  if(iflag.eq.3)then
                     if(norder.gt.0)then
                        call gmprd(p,derv,derw,nobs,nobs,1)
                     else
                        call mcpy(derv,derw,nobs,1,0)
                     endif
                     call stormat(derw,dwmat,n,nobs,np,kpar)
                  endif
c Compute gradlamy from gradlamx(.) if the estimated LAMBDA(Y) is set equal to
c one of the estimated LAMBDA(X)'s.
                  if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.i)then
                     gradlamx(i)=gradlamx(i)+sumlny
                     gradlamy=gradlamx(i)
                  endif
                  grad(kpar)=-gradlamx(i)
               endif
   70       continue
         endif
      endif

c*********************************************************************
c At the INITIAL STEP (IFLAG = 1), only BETA and SIGW2 are estimated
c conditionally on the initial values of RHO's, PI's, DELTA's, LAMBDA(Z)'s,
```

```
c LAMBDA(Y) and LAMBDA(X)'s, hence only the estimates of BETA and SIGW2
c and their associated statistics (standard errors, t-students, first
c derivatives of L) are computed.
c***********************************************************************
      if(iflag.eq.1)then
c Compute the covariance matrix of BETA which will be stored in <xppx>.
         call smpy(xppx,sigw2,xppx,nb,nb,1)
c Compute the gradient of BETA: <gradbeta>.
         onesigw2=1./sigw2
         call gtprd(px,w,gradbeta,nobs,nb,1)
         call smpy(gradbeta,onesigw2,gradbeta,nb,1,0)
c Compute the derivatives and elasticities of Y (sample)
c with respect to X and Z, at the sample means of Y, X and Z.
c SEPT 99: Compute the Marginal Rate of Substitution between two independent
c          variables, using the ratio of these derivatives.
         call smpelast(ymean,xmean,zmean,beta,delta,vly,vlx,vlz,
     +   nx,nobs,nullx,nb,nvaz,nlamy,numlx,numlz,numxeqz,namx,
     +   smpderx,smpelax,smpderz,smpelaz)
c Compute the expected value of Y (expy = E(Y)) for each observation in the
c sample, and the average values of the derivatives and elasticities of E(Y)
c with respect to X and Z.
c SEPT 99: Replace the old definition of "r2e" with the new one (Pearson R2:
c          square of the coefficient of correlation between the observed Y
c          and the expected value of Y).
         call expelast(y,x,xbox,z,p,v,sqrtfz,beta,delta,vly,vlx,vlz,
     +   sigw2,n,nx,nz,nobs,nullx,nb,nvaz,nlamy,numlx,numlz,numxeqz,
     +   expy,expderx,expelax,expderz,expelaz,ysam,r2e,plim,eymean,
     +   nexpy,iflag)
c Print BETA, SIGW2 and their associated statistics.
c Sept 10 2001
c Initial step: if npar=0, i.e. no other parameters than the beta's are
c               estimated, write the results in initial step (iflag=1)
c               on unit=1, file=TabFile for TABLEX.
         if(npar.eq.0)then
            open(unit=1,file=TabFile)
            write(1,1415)namy,varname,nvariant
            write(1,1416)nobs,nvaxz,nvax,norder,nvaz,nlamz,nlamy,nlamx
         endif
         if(noprint.eq.0)then
            print 1080
            print 1090
         endif
         do 71 i=1,nb
            call loc(i,i,ir,nb,nb,1)
            stderr=dsqrt(xppx(ir))
            student=beta(i)/stderr
            ntypelx='nl'
            if(numlx(i).gt.0)then
               k=numlx(i)
               ntypelx='fl'
               if(ntlx(k).eq.1)ntypelx='el'
            endif
            if(npar.eq.0)then
                  write(1,1420)namx(i),numlx(i),beta(i),ntypelx,
     +            student,student,coryx(i,1),coryx(i,1),
     +            smpderx(i),smpelax(i),expderx(i),expelax(i)
c September 2008 for TABQDF
```

```
                  if(namx(i).ne.'CONSTANT'.and.namx(i).ne.'constant'
     +.and.namx(i).ne.'Constant')write(4,1403)namx(i),student,smpelax(i)
1403              format(a10,2d15.7)
              endif
            if(noprint.eq.0)print 1100,namx(i),numlx(i),beta(i),stderr,
     +        student,gradbeta(i),smpderx(i),smpelax(i),expderx(i),
     +      expelax(i)
            stderr=sqrt(2./nobs)*sigw2
            student=sigw2/stderr
   71    continue
c By definition, the first derivative of the log-likelihood with respect to
c SIGW2 is exactly equal to zero at the maximum of the function, hence it needs
c not be computed.
         grsigw2=0.d0
      if(noprint.eq.0)then
         print 1110,sigw2,stderr,student,grsigw2
         print 1120
      endif

c Print the initial values of RHO's, PI's.
         if(norder.gt.0)then
      if(noprint.eq.0)then
            print 1130
            do 72 i=1,norder
   72       print 1140,i,rho(i)
            do 73 i=1,norder
   73       print 1150,i,pi(i)
      endif
         endif
c Print the initial values of DELTA's and LAMBDA(Z)'s.
c July 3, 2003: store the correlations between Original Y and Z_m in coryz(.,1)
c where the second index is 1 when Y and Z-m are in original units.
         if(nvaz.gt.0)then
            if(noprint.eq.0)print 1130
            student=0.d0
            do 74 i=1,nvaz
            if(npar.eq.0)write(1,1430)namz(i),numlz(i),delta(i),
     +        ntdel(i),student,student,coryz(i,1),coryz(i,1),
     +      smpderz(i),smpelaz(i),expderz(i),expelaz(i)
   74       if(noprint.eq.0)print 1160,namz(i),numlz(i),delta(i)
            if(nlamz.gt.0)then
               do 75 i=1,nlamz
               if(npar.eq.0)write(1,1440)i,vlz(i),ntlz(i)
   75          if(noprint.eq.0)print 1170,i,vlz(i)
            endif
         endif
c Print the initial values of LAMBDA(Y) and LAMBDA(X)'s.
         if(noprint.eq.0)print 1130
         if(nlamy.eq.1)then
            if(noprint.eq.0)print 1180,vly
            if(npar.eq.0)then
               student=0.
               stud1=0.
                 write(1,1450)numlxy,vly,ntly,student,stud1
            endif
         endif
         if(nlamx.gt.0)then
```

```
          do 76 i=1,nlamx
          if(npar.eq.0)then
             student=0.
             stud1=0.
               write(1,1460)i,vlx(i),ntlx(i),student,stud1
          endif
   76     if(noprint.eq.0)print 1190,i,vlx(i)
        endif
        if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.gt.0)then
          if(noprint.eq.0)print 1200,numlxy
        endif
c Print Log-likelihood value in terms of maximum.
        flik=-fun
      if(noprint.eq.0)then
        print 1210,flik
        print 1211,nobs
      endif
c Compute R2L = Pseudo-R2(L) (unadjusted)
c and  R2LBAR = Pseudo-R2(L) (adjusted for degrees of freedom)
        pymean=0.d0
        do 110 i=1,nobs
  110   pymean=pymean+py(i)/nobs
        sigw2b0=0.d0
        do 77 i=1,nobs
   77   sigw2b0=sigw2b0+(py(i)-pymean)**2
        sigw2b0=sigw2b0/nobs
        flikb0=-conlik-nobs*dlog(sigw2b0)/2
        if(norder.gt.0)flikb0=flikb0+dlog(dabs(detp))
        if(nvaz.gt.0)flikb0=flikb0-sumlnfz/2
        if(nlamy.eq.1)flikb0=flikb0+(vly-1)*sumlny
        r2ebar=1-(nobs-1)*(1-r2e)/(nobs-nb)
        r2l=1-dexp(2*(flikb0-flik)/nobs)
        r2lbar=1-(nobs-1)*(1-r2l)/(nobs-nb)
c September 29 2008
c Compute 95% Confidence Interval (wlower,wupper) for Residual W
        temp=1.96*dsqrt(sigw2)
        wlower=wmean-temp
        wupper=wmean+temp
      if(noprint.eq.0)then
        print 1230,r2e,r2ebar
        print 1220,r2l,r2lbar
        print 1240,plim,eymean
c Print Mean(W), Variance(W) and Confidence Interval for W
        print 1241,wmean,sigw2,wlower,wupper
 1241   format(/'MEAN of residual W                  =',d15.7,
     +         /'VARIANCE of residual W              =',d15.7,
     +         /'95% CONFIDENCE INTERVAL for residual W'
     +         /'Lower Bound =',d15.7,3x,'Upper Bound =',d15.7)
      endif
        if(npar.eq.0)then
          write(1,1480)flik,r2e,r2ebar,r2l,r2lbar,plim
          write(1,1481)ymean,eymean
        endif
      if(noprint.eq.0)then
        if(nres.gt.0)then
c 29/8/94: Print residuals U, V and W.
          print*,'Residuals U,V,W'
```

```
          do 79 i=1,nres
   79     print 1410,i,u(i),v(i),w(i)
        endif
      endif
        close(1)
c If there is no extra parameter (npar=0) to estimate, i.e. apart from BETA and
SIGW2,
c the program will stop, since there is no need to maximize the concentrated
c log-likelihood function which depends only on the extra parameters.
        if(npar.eq.0)then
             call cputime(2)
             stop
        endif
      endif
c END of INITIAL STEP (IFLAG=1)

c*************************************************************************
c At the FINAL STEP (IFLAG=3), print the parameter estimates of BETA's,
c VAR(W), RHO's, PI's, DELTA's, LAMBDA(Z)'s, LAMBDA(Y) and LAMBDA(X)'s,
c and their associated statistics (standard errors, t-students, and first
c derivatives of L).
c*************************************************************************
      if(iflag.eq.3)then
c Compute the covariance matrices COV and CONDCOV by BHHH method.
c NB1 = number of estimated BETA's and SIGW2.
c NP1 = number of estimated BETA's, SIGW2, RHO's, PI's, DELTA's, LAMBDA(Z)'s
c LAMBDA(Y) and LAMBDA(X)'s, associated with COV.
c NP2 = number of estimated BETA's, SIGW2, RHO's, PI's and DELTA's associated
c with CONDCOV.
c NPP1= number of distinct elements in the (NP1 x NP1) symmetric covariance
c       matrix COV, stored columnwise as an upper triangular matrix.
        nb1=nb+1
        np1=nb1+npar
        np2=np1
        if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.0)np2=np2-1
        if(nlamx.gt.0)np2=np2-nestlamx
        if(nlamz.gt.0)np2=np2-nestlamz
        npp1=(np1*(np1+1))/2
        onesigw2=1./sigw2
        do 80 i=1,npp1
   80   cov(i)=0.d0
c Compute the derivative DERLT of the nonconcentrated log-likelihood function
c associated with each observation t.
c Convert PX from single to double dimension array so that the elements of the
c matrix can be located with two indices (row and column) while computing DERLT
c only.
        call array(1,nobs,nb,n,nx,px,px)
        do 89 i=1,nobs
          ws=w(i)/sigw2
c DERLT with respect to BETA's.
          do 81 j=1,nb
   81     derlt(j)=ws*px(i,j)
c DERLT with respect to SIGW2.
          jpar=nb1
          derlt(jpar)=(ws*ws-onesigw2)/2
c DERLT with respect to RHO's.
          kpar=0
```

```
          if(norder.gt.0)then
             do 82 k=1,norder
                jpar=jpar+1
                kpar=kpar+1
   82        derlt(jpar)=-ws*dwmat(i,kpar) + tracerho(k)/nobs
c DERLT with respect to PI's.
             do 83 k=1,norder
                if(ntpi(k).eq.1)then
                   jpar=jpar+1
                   kpar=kpar+1
                   derlt(jpar)=-ws*dwmat(i,kpar) + tracepi(k)/nobs
                endif
   83        continue
          endif
c DERLT with respect to DELTA's.
          if(nestdelta.gt.0)then
             do 84 k=1,nvaz
                if(ntdel(k).eq.1)then
                   jpar=jpar+1
                   kpar=kpar+1
                   derlt(jpar)=-ws*dwmat(i,kpar)-zbox(i,k)/2
                endif
   84        continue
          endif
c DERLT with respect to LAMBDA(Z)'s.
          if(nestlamz.gt.0)then
             do 86 k=1,nlamz
                if(ntlz(k).eq.1)then
                   jpar=jpar+1
                   kpar=kpar+1
                   derlt(jpar)=-ws*dwmat(i,kpar)
                   do 85 j=1,nvaz
                      if(numlz(j).eq.k) derlt(jpar)=derlt(jpar)-delta
     +                (j)*derzl(z(i,j),vlz(k))/2
   85              continue
                endif
   86        continue
          endif
c DERLT with respect to LAMBDA(Y).
          if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.0)then
             jpar=jpar+1
             kpar=kpar+1
             derlt(jpar)=-ws*dwmat(i,kpar) + dlog(y(i))
          endif
c DERLT with respect to LAMBDA(X)'s.
          if(nestlamx.gt.0)then
             do 87 k=1,nlamx
                if(ntlx(k).eq.1)then
                   jpar=jpar+1
                   kpar=kpar+1
                   derlt(jpar)=-ws*dwmat(i,kpar)
                   if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.eq.k) derlt
     +             (jpar)=derlt(jpar)+dlog(y(i))
                endif
   87        continue
          endif
c Sum the cross products of DERLT's over NOBS observations.
```

```
c This sum is stored in COV.
            do 88 k=1,np1
               do 88 m=1,k
                  call loc(k,m,km,np1,np1,1)
   88       cov(km)=cov(km)+derlt(k)*derlt(m)
   89    continue
c After computing DERLT, reconvert PX from double to single dimension array
c so that the matrix can be used in the subroutines which work only with
c single dimension arrays.
         call array(2,nobs,nb,n,nx,px,px)

c Create CONDCOV from the first NP2 rows and the first NP2 columns of COV.
         do 90 k=1,np2
            do 90 m=1,k
               call loc(k,m,km,np1,np1,1)
   90    condcov(km)=cov(km)
c Invert COV to obtain the covariance matrix of all the parameter estimates.
         call sinv(cov,np1,epsinv,ier)
         if(ier.ne.0)then
            if(noprint.eq.0)print*,'Matrix COV not invertible ...'
            stop 777
         endif
c Invert CONDCOV to obtain the covariance matrix of the estimates of BETA,
c SIGW2, RHO, PI and DELTA conditional on the values (fixed or estimated)
c of LAMBDA(Z), LAMBDA(Y) and LAMBDA(Z).
         if(np2.lt.np1)then
            call sinv(condcov,np2,epsinv,ier)
            if(ier.ne.0)then
               if(noprint.eq.0)print*,'Matrix CONDCOV not invertible ...
     +           '
               stop 7777
            endif
         else
            call mcpy(cov,condcov,np2,np2,1)
         endif

c APRIL 1998
c Compute the Table of Variance-Decomposition Proportions for the Transformed
c        Independent Variables X (including the Constant)
c        and the Correlation Matrix for the Transformed Independent Variables X
c        (excluding the Constant) and the Dependent Variable Y.
c XTMEAN : Vector of the Means of the Transformed Independent Variables,
c          except that the last variable which corresponds to the transformed
c          constant is replaced by the transformed dependent variable, before
c          calling the subroutine CORREL.
c XTCONST: Work Vector containing the Transformed Constant.
         if(ncor.eq.1.or.ncor.eq.2)call vardec(px,xppx,namx,n,nobs,
     +   nb,nxx)
         do 91 i=1,nobs
         xtconst(i)=px(i,nb)
   91    px(i,nb)=py(i)
         do 92 j=1,nb
         xtmean(j)=0.
         do 92 i=1,nobs
   92    xtmean(j)=xtmean(j)+px(i,j)/nobs
         call correl(px,xtmean,xppy,xppx,namx,namy,n,nobs,nb,nxx,
     +   ncor)
```

```
        do 93 i=1,nobs
   93   px(i,nb)=xtconst(i)
        nbb=(nb*(nb-1))/2
        do 105 i=1,nb-1
  105   coryx(i,2)=xppx(nbb+i)
        coryx(nb,2)=0.d0
c JULY 24 2003
c Compute the correlations between Box-Cox Transformed Y and Z_m (m=1,nvaz):
c coryz(i,2),i=1,nvazy
      if(nvaz.gt.0)then
        nvazy=nvaz+1
        do 133 j=1,nvazy
        ztmean(j)=0.
        do 133 i=1,nobs
        if(j.lt.nvazy)then
          pz(i,j)=zbox(i,j)
        else
          pz(i,j)=ybox(i)
        endif
  133   ztmean(j)=ztmean(j)+pz(i,j)/nobs
        call correl(pz,ztmean,xppy,xppx,namz,namy,n,nobs,nvazy,nzz,ncor)
        nbb=(nvazy*(nvazy-1))/2
        do 134 i=1,nvaz
  134   coryz(i,2)=xppx(nbb+i)
      endif


c Compute the gradient of BETA: GRADBETA.
        call gtprd(px,w,gradbeta,nobs,nb,1)
        call smpy(gradbeta,onesigw2,gradbeta,nb,1,0)
c Compute the derivatives and elasticities of Y (sample)

c with respect to X and Z, at the sample means of Y, X and Z.
c SEPT 99: Compute the Marginal Rate of Substitution (MRS) between two
c          independent variables, using the ratio of these derivatives.
c          This MRS is computed in Subroutine smpelast.

        call smpelast(ymean,xmean,zmean,beta,delta,vly,vlx,vlz,
     +  nx,nobs,nullx,nb,nvaz,nlamy,numlx,numlz,numxeqz,namx,
     +  smpderx,smpelax,smpderz,smpelaz)

c Compute the expected value of Y (expy = E(Y)) for each observation in the
c sample, and the average values of the derivatives and elasticities of E(Y)
c with respect to X and Z.
c  SEPT 99:Replace the old definition of "r2e" with the Pearson R2(E) (square of
c          the coefficient of correlation between the observed Y and the
c          expected value E(Y)). This R2(E) is computed in Subroutine EXPELAST.
        call expelast(y,x,xbox,z,p,v,sqrtfz,beta,delta,vly,vlx,vlz,
     +  sigw2,n,nx,nz,nobs,nullx,nb,nvaz,nlamy,numlx,numlz,numxeqz,
     +  expy,expderx,expelax,expderz,expelaz,ysam,r2e,plim,eymean,
     +  nexpy,iflag)
c Print BETA and its associated statistics.
c Write the results of final step (iflag=3) on unit=1, file=TabFile for TABLEX.
        open(unit=1,file=TabFile)
        write(1,1415)namy,varname,nvariant
        write(1,1416)nobs,nvaxz,nvax,norder,nvaz,nlamz,nlamy,nlamx
      if(noprint.eq.0)then
```

```
         print 1250
         print 1260
         print 1270
      endif
         do 94 i=1,nb
            ir=(i*(i+1))/2
            stderr=dsqrt(cov(ir))
            student=beta(i)/stderr
            if(noprint.eq.0)print 1280,namx(i),numlx(i),beta(i),stderr,
     +        student,gradbeta(i), smpderx(i),smpelax(i)
            stdcond=dsqrt(condcov(ir))
            stucond=beta(i)/stdcond
c September 2008 for TABQDF
              if(namx(i).ne.'CONSTANT'.and.namx(i).ne.'constant'
     +.and.namx(i).ne.'Constant')write(4,1403)namx(i),stucond,smpelax(i)

            if(noprint.eq.0)print 1290,stdcond,stucond,expderx(i),
     +      expelax(i)
            ntypelx='nl'
            if(numlx(i).gt.0)then
               k=numlx(i)
               ntypelx='fl'
               if(ntlx(k).eq.1)ntypelx='el'
            endif
            write(1,1420)namx(i),numlx(i),beta(i),ntypelx,student,
     +      stucond,coryx(i,1),coryx(i,2),smpderx(i),smpelax(i),
     +      expderx(i),expelax(i)
   94    continue

c Print SIGW2 and its associated statistics.
         ir=(nb1*(nb1+1))/2
         stderr=dsqrt(cov(ir))
         student=sigw2/stderr
c By definition, the first derivative of the log-likelihood with respect to
c SIGW2 is exactly equal to zero at the maximum of the function, hence it needs
c not be computed.
         grsigw2=0.d0
         if(noprint.eq.0)print 1300,sigw2,stderr,student,grsigw2
         stdcond=dsqrt(condcov(ir))
         stucond=sigw2/stdcond
         if(noprint.eq.0)print 1290,stdcond,stucond

         jpar=nb1
         if(norder.gt.0)then
            if(noprint.eq.0)print 1130
                   do 194 i=1,norder
  194          write(1,1001)matname(i)
            do 95 i=1,norder
c Print RHO and its associated statistics.
c RHO is always estimated (E).
               jpar=jpar+1
               ir=(jpar*(jpar+1))/2
               stderr=dsqrt(cov(ir))
               student=rho(i)/stderr
               status='(E)'
               if(noprint.eq.0)print 1310,i,status,rho(i),stderr,student
     +             ,gradrho(i)
```

```
              stdcond=dsqrt(condcov(ir))
              stucond=rho(i)/stdcond
              if(noprint.eq.0)print 1290,stdcond,stucond
              write(1,1292)i,rho(i),student,stucond
   95        continue
c Print PI and its associated statistics.
c PI can be fixed (F) or estimated (E).
                  do 195 i=1,norder
              if(ntpi(i).eq.0)then
                 student=0.d0
                       stud1=0.d0
                       stucond=0.d0
                       stu1cond=0.d0
                 status='(F)'
                 if(noprint.eq.0)print 1320,i,status,pi(i)
              else
                 jpar=jpar+1
                 ir=(jpar*(jpar+1))/2
                 stderr=dsqrt(cov(ir))
                 student=pi(i)/stderr
                 stud1=(pi(i)-1.)/stderr
                 status='(E)'
                 if(noprint.eq.0)print 1320,i,status,pi(i),stderr,
     +                 student,stud1,gradpi(i)
                 stdcond=dsqrt(condcov(ir))
                 stucond=pi(i)/stdcond
                 stu1cond=(pi(i)-1.)/stdcond
                 if(noprint.eq.0)print 1330,stdcond,stucond,stu1cond
              endif
              write(1,1293)i,pi(i),ntpi(i),student,stud1,stucond,
     +        stu1cond
  195             continue

        endif

c Print DELTA and its associated statistics.
c DELTA can be fixed (F) or estimated (E).
        if(nvaz.gt.0)then
           if(noprint.eq.0)print 1130
           do 97 i=1,nvaz
              if(ntdel(i).eq.0)then
                 student=0.d0
                       stucond=0.d0
                 status='(F)'
                 if(noprint.eq.0)then
                       print 1340,namz(i),status,numlz(i),
     +           delta(i),smpderz(i),smpelaz(i)
                 print 1350,expderz(i),expelaz(i)
                 endif
              else
                 jpar=jpar+1
                 ir=(jpar*(jpar+1))/2
                 stderr=dsqrt(cov(ir))
                 student=delta(i)/stderr
                 status='(E)'
                 if(noprint.eq.0)print 1360,namz(i),status,numlz(i),
     +           delta(i),stderr,student, graddelta(i),smpderz(i),
```

```
     +            smpelaz(i)
                  stdcond=dsqrt(condcov(ir))
                  stucond=delta(i)/stdcond
                  if(noprint.eq.0)print 1290,stdcond,stucond,expderz(i),
     +                  expelaz(i)
               endif
               write(1,1430)namz(i),numlz(i),delta(i),ntdel(i),
     +         student,stucond,coryz(i,1),coryz(i,2),
     +         smpderz(i),smpelaz(i),expderz(i),expelaz(i)
   97       continue

c Print LAMBDA(Z) and its associated statistics.
c LAMBDA(Z) can be fixed (F) or estimated (E).
            if(nlamz.gt.0)then
               do 98 i=1,nlamz
                  if(ntlz(i).eq.0)then
                     student=0.d0
                           stud1=0.d0
                     status='(F)'
                     if(noprint.eq.0)print 1370,i,status,vlz(i)
                  else
                     jpar=jpar+1
                     ir=(jpar*(jpar+1))/2
                     stderr=dsqrt(cov(ir))
                     student=vlz(i)/stderr
                     stud1=(vlz(i)-1.)/stderr
                     status='(E)'
                     if(noprint.eq.0)print 1370,i,status,vlz(i),stderr,
     +               student,stud1,gradlamz(i)
                  endif
                  write(1,1440)i,vlz(i),ntlz(i),student,stud1
   98          continue
            endif
         endif

c Print LAMBDA(Y) and its associated statistics.
c LAMBDA(Y) can be fixed (F) or estimated (E).
         if(noprint.eq.0)print 1130
         if(nlamy.eq.1)then
            if(ntly.eq.0)then
               student=0.d0
                   stud1=0.d0
               status='(F)'
               if(noprint.eq.0)print 1380,status,vly
            else
               if(numlxy.eq.0)then
                  jpar=jpar+1
                  ir=(jpar*(jpar+1))/2
               else
                  k=0
                  do 99 i=1,nlamx
                     if(ntlx(i).eq.1)then
                        k=k+1
                        if(numlxy.eq.i)ilamx=k
                     endif
   99             continue
                  j=jpar+ilamx
```

```
                  ir=(j*(j+1))/2
               endif
               stderr=dsqrt(cov(ir))
               student=vly/stderr
               stud1=(vly-1.)/stderr
               status='(E)'
               if(noprint.eq.0)print 1380,status,vly,stderr,student,
     +         stud1,gradlamy
            endif
            write(1,1450)numlxy,vly,ntly,student,stud1
         endif

c Print LAMBDA(X) and its associated statistics.
c LAMBDA(X) can be fixed (F) or estimated (E).
         if(nlamx.gt.0)then
            do 100 i=1,nlamx
               if(ntlx(i).eq.0)then
                  student=0.d0
                        stud1=0.d0
                  status='(F)'
                  if(noprint.eq.0)print 1390,i,status,vlx(i)
               else
                  jpar=jpar+1
                  ir=(jpar*(jpar+1))/2
                  stderr=dsqrt(cov(ir))
                  student=vlx(i)/stderr
                  stud1=(vlx(i)-1.)/stderr
                  status='(E)'
                  if(noprint.eq.0)print 1390,i,status,vlx(i),stderr,
     +            student,stud1,gradlamx(i)
               endif
               write(1,1460)i,vlx(i),ntlx(i),student,stud1
  100       continue
         endif
         flik=-fun
         if(nlamy.eq.1.and.ntly.eq.1.and.numlxy.gt.0)then
              if(noprint.eq.0)print 1200,numlxy
         endif
         if(noprint.eq.0)then
         print 1210,flik
         print 1211,nobs
         endif
c 22 June 2000: Compute R2EBAR = Pearson R2(E) adjusted for degrees of freedom.
         r2ebar=1-(nobs-1)*(1-r2e)/(nobs-(np1-1))
         if(noprint.eq.0)print 1230,r2e,r2ebar
c Compute R2L = Pseudo-R2(L) (unadjusted)
c and  R2LBAR = Pseudo-R2(L) (adjusted for degrees of freedom)
         pymean=0.d0
         do 120 i=1,nobs
  120    pymean=pymean+py(i)/nobs
         sigw2b0=0.d0
         do 1011 i=1,nobs
 1011    sigw2b0=sigw2b0+(py(i)-pymean)**2
         sigw2b0=sigw2b0/nobs
         flikb0=-conlik-nobs*dlog(sigw2b0)/2
         if(norder.gt.0)flikb0=flikb0+dlog(dabs(detp))
         if(nvaz.gt.0)flikb0=flikb0-sumlnfz/2
```

```
         if(nlamy.eq.1)flikb0=flikb0+(vly-1)*sumlny

         r2l=1-dexp(2*(flikb0-flik)/nobs)
         r2lbar=1-(nobs-1)*(1-r2l)/(nobs-(np1-1))
c September 29 2008
c Compute Mean of Residual W: wmean
            wmean=0.d0
            do 157 i=1,nobs
  157       wmean=wmean+w(i)/nobs
c Compute 95% Confidence Interval (wlower,wupper) for Residual W
         temp=1.96*dsqrt(sigw2)
         wlower=wmean-temp
         wupper=wmean+temp
      if(noprint.eq.0)then
         print 1220,r2l,r2lbar
         print 1240,plim,eymean
c Print Mean(W), Variance(W) and Confidence Interval for W
         print 1241,wmean,sigw2,wlower,wupper
      endif
         write(1,1480)flik,r2e,r2ebar,r2l,r2lbar,plim
         write(1,1481)ymean,eymean
         close(1)
         close(4)
      if(noprint.eq.0)then
         if(nres.gt.0)then
c 29/8/94: Print residuals U, V and W.
            print*,'Residuals U,V,W'
            do 1013 i=1,nres
 1013       print 1410,i,u(i),v(i),w(i)
         endif
      endif
      endif

      return
 1001 format(a8)
 1010 format(20f3.0)
 1020 format(a10)
 1030 format(/'VARIABLE',t16,'MINIMUM',t32,'MEAN',t46,'MAXIMUM', t56,
     +'NULL OBS.',t71,'SCALE')
 1040 format('DEPENDENT'/a10,3d15.7,3x,i3,3x,d15.7)
 1050 format('INDEPENDENT')
 1060 format(a10,3d15.7,3x,i3,3x,d15.7)
 1070 format('HETEROSKEDASTICITY')
 1080 format(//'INITIAL ESTIMATES')
 1090 format(/t14,'LAM',t21,'PARAMETER',t36,'STD-ERROR', t53,'STUDENT-T'
     +,t66,'GRADIENT',t81,'DERIV.Y',t97,'ELAST.Y' /t81,'DERIV.E(Y)',t97,
     +'ELAST.E(Y)'/)
 1100 format(/a10,4x,'X',i1,2d15.7,f15.7,2d15.7,f15.7/76x,d15.7, f15.7)

 1110 format(/'VAR(W)',10x,2d15.7,f15.7,d15.7)
 1120 format(/'INITIAL VALUES')
 1130 format(1h )
 1140 format('RHO',i1,9x,d15.7)
 1150 format('PI',i1,10x,d15.7)
 1160 format(a10,1x,'Z',i1,d15.7)
 1170 format('LAM(Z',i1,')',6x,d15.7)
 1180 format('LAM(Y)',7x,d15.7)
```

```
 1190 format('LAM(X',i1,')',6x,d15.7)
 1200 format(/'LAM(Y) is set equal to LAM(X'i1').')
 1210 format(//'LOG-LIKELIHOOD =',d15.7)
 1211 format(/'NUMBER OF OBSERVATIONS              =',7x,i8)
 1220 format(/'PSEUDO-R2(L)  (unadjusted)          =',f15.4,
     +       /'PSEUDO-R2(L)  (adjusted for D.F.)   =',f15.4)
 1230 format(/'PEARSON-R2(E) (unadjusted)          =',f15.4,
     +       /'PEARSON-R2(E) (adjusted for D.F.)   =',f15.4)
 1240 format(/'PROBABILITY of Y to be at the limit =',f15.4,
     +       /'MEAN VALUE of E(Y)                  =',d15.7)
 1250 format(//'FINAL ESTIMATES')
 1260 format(/t14,'LAM',t21,'PARAMETER',t36,'STD-ERROR', t53,'STUDENT-T'
     +,t68,'STUDENT-T',t81,'GRADIENT', t96,'DERIV.Y',t112,'ELAST.Y',
     +/t36,'(CONDIT.)',t53,'(CONDIT.)',t68,'(CONDIT.)', t96,'DERIV.E(Y)'
     +,t112,'ELAST.E(Y)')

 1270 format(t53,'(PAR = 0)',t68,'(PAR = 1)')
 1280 format(/a10,4x,'X',i1,2d15.7,f15.7,15x,2d15.7,f15.7)
 1290 format(31x,d15.7,f15.7,30x,d15.7,f15.7)
 1292 format('rho',i1,d20.12,3x,2f12.4)
 1293 format('pi ',i1,d20.12,i3,4f12.4)
 1300 format(/'VAR(W)',10x,2d15.7,f15.7,15x,d15.7)
 1310 format('RHO'i1,6x,a3,3x,2d15.7,f15.7,15x,d15.7)
 1320 format('PI'i1,7x,a3,3x,2d15.7,2f15.7,d15.7)
 1330 format(31x,d15.7,2f15.7)
 1340 format(a10,a3,1x,'Z',i1,d15.7,60x,d15.7,f15.7)
 1350 format(91x,d15.7,f15.7)
 1360 format(a10,a3,1x,'Z',i1,2d15.7,f15.7,15x,2d15.7,f15.7)
 1370 format('LAM(Z',i1,')',3x,a3,3x,2d15.7,2f15.7,d15.7)
 1380 format('LAM(Y)',4x,a3,3x,2d15.7,2f15.7,d15.7)
 1390 format('LAM(X'i1')',3x,a3,3x,2d15.7,2f15.7,d15.7)
 1410 format(i4,3d15.7)
 1415 format(a8,2x,a9,i4)
 1416 format(8i5)
 1420 format(a8,2x,i2,d20.12,a3,4f12.4/4d20.12)
 1430 format(a8,2x,i2,d20.12,i3,4f12.4/4d20.12)
 1440 format('lambda(z)',1x,i2,d20.12,i3,2f12.4)
 1450 format('lambda(y)',1x,i2,d20.12,i3,2f12.4)
 1460 format('lambda(x)',1x,i2,d20.12,i3,2f12.4)
 1480 format(6f12.3)
 1481 format(2d20.12)
      end

      subroutine gmadd(a,b,r,n,m)
c Add two general matrices A and B to form resultant matrix R.
c Arguments:
c   A - Name of first input matrix.
c   B - Name of second input matrix.
c   R - Name of output matrix.
c   N - Number of rows in A, B, R.
c   M - Number of columns in A, B, R.
c Remarks:
c   All matrices must be stored as general matrices.
      double precision a(*),b(*),r(*)
      nm=n*m
      do 1 i=1,nm
1     r(i)=a(i)+b(i)
```

```
      return
      end

c
c     ..................................................................
c
c        subroutine gmprd
c
c        purpose
c           multiply two general matrices to form a resultant general
c           matrix
c
c        usage
c           call gmprd(a,b,r,n,m,l)
c
c        description of parameters
c           a - name of first input matrix
c           b - name of second input matrix
c           r - name of output matrix
c           n - number of rows in a
c           m - number of columns in a and rows in b
c           l - number of columns in b
c
c        remarks
c           all matrices must be stored as general matrices
c           matrix r cannot be in the same location as matrix a
c           matrix r cannot be in the same location as matrix b
c           number of columns of matrix a must be equal to number of row
c           of matrix b
c
c        subroutines and function subprograms required
c           none
c
c        method
c           the m by l matrix b is premultiplied by the n by m matrix a
c           and the result is stored in the n by l matrix r.
c
c     ..................................................................
c
      subroutine gmprd(a,b,r,n,m,l)
      implicit double precision (a-h,o-z)
      dimension a(1),b(1),r(1)
c
      ir=0
      ik=-m
      do 10 k=1,l
      ik=ik+m
      do 10 j=1,n
      ir=ir+1
      ji=j-n
      ib=ik
      r(ir)=0
      do 10 i=1,m
      ji=ji+n
      ib=ib+1
   10 r(ir)=r(ir)+a(ji)*b(ib)
      return
```

```
      end

c
c     ..................................................................
c
c        subroutine gmsub
c
c        purpose
c           subtract one general matrix from another to form resultant
c           matrix
c
c        usage
c           call gmsub(a,b,r,n,m)
c
c        description of parameters
c           a - name of first input matrix
c           b - name of second input matrix
c           r - name of output matrix
c           n - number of rows in a,b,r
c           m - number of columns in a,b,r
c
c        remarks
c           all matrices must be stored as general matrices
c
c        subroutines and function subprograms required
c           none
c
c        method
c           matrix b elements are subtracted from corresponding matrix a
c           elements
c
c     ..................................................................
c
      subroutine gmsub(a,b,r,n,m)
      implicit double precision (a-h,o-z)
      dimension a(1),b(1),r(1)
c
c        calculate number of elements
c
      nm=n*m
c
c        subtract matrices
c
      do 10 i=1,nm
   10 r(i)=a(i)-b(i)
      return
      end

c
c     ..................................................................
c
c        subroutine gtprd
c
c        purpose
c           premultiply a general matrix by the transpose of another
c           general matrix
c
```

```
c        usage
c           call gtprd(a,b,r,n,m,l)
c
c        description of parameters
c           a - name of first input matrix
c           b - name of second input matrix
c           r - name of output matrix
c           n - number of rows in a and b
c           m - number of columns in a and rows in r
c           l - number of columns in b and r
c
c        remarks
c           matrix r cannot be in the same location as matrix a
c           matrix r cannot be in the same location as matrix b
c           all matrices must be stored as general matrices
c
c        subroutines and function subprograms required
c           none
c
c        method
c           matrix transpose of a is not actually calculated. instead,
c           elements of matrix a are taken columnwise rather than
c           rowwise for postmultiplication by matrix b.
c
c     ..................................................................
c
      subroutine gtprd(a,b,r,n,m,l)
      implicit double precision (a-h,o-z)
      dimension a(1),b(1),r(1)
c
      ir=0
      ik=-n
      do 10 k=1,l
      ij=0
      ik=ik+n
      do 10 j=1,m
      ib=ik
      ir=ir+1
      r(ir)=0.
      do 10 i=1,n
      ij=ij+1
      ib=ib+1
   10 r(ir)=r(ir)+a(ij)*b(ib)
      return
      end

      program Level2
c*******************************************************************************
c     PROGRAM LEVEL2: compiled with Compaq Visual Fortran 6.6 (2008)
c
c     PUBLICATION CRT-972:
c     LEVEL: The L-2.0 program for BC-DAUHESEQ regressions --- Box-Cox Directed
c     AUtoregressive HEteroskedastic Single Equation models.
c        by TRAN LIEM, MARC GAUDRY and ULRICH BLUM.
c     Universite de Montreal, Centre de recherche sur les transports,
c     June 1998.
c*******************************************************************************
```

```
c   The main program contains two minimization algorithms :
c   1- r.fletcher and m.j.d.powell,"a rapidly convergent descent method
c      for minimization", computer journal,6,pp.163,168 (1963),
c   2- m.j.d.powell,"an efficient method for finding the minimum of a
c      function of several variables without calculating derivatives",
c      computer journal,7,pp.155,162 (1964).
c*****************************************************************************
      implicit double precision(a-h,o-z)
      real*4 epsi,wght
      logical dercal
c Parameters used in Dimension:
c n  - maximum number of parameters estimated in log-likelihood function:
c      n = 2 rho's + 2 pi's + 1 lambda(y) + 10 lambda(x)
c          + 10 delta's + 10 lambda(z) = 35 parameters.
c n1 = n + 1.
      parameter (n=35,n1=n+1)
      dimension d(n),y(n),x(n),g(n),p(n)
      dimension et(n,n),wt(n),s(n1),dirin(n1),xs(n1,n)

      dimension isw(6),e(2)
      common nlist,nstep,epsi,wght,noprint
c noprint = 1 don't print the ouput.
c         = 0 print the output.
      data isw/6*0/
      data epsilon/1.d-6/
   44 format(//'the run has been terminated since no improvement can be
     1found on the current minimum,'/' the parameters of which follow
     2- - f,x(i)')
  105 format (4(/),'initialization cards for control routine follow - -
     1 id card - no. parameters , no. steps , step size ',/,'minimizatio
     2n tolerance - starting values of parameters - relative uncertainti
     3es of parameters (error estimates) -  ',/,'and starting direction
     4 (relative parameter increments for 1st step)')
  110 format ('identification line')
  120 format(2i10,2e10.1)
  130 format (10e13.5)
  280 format (/'first entry to fcn , parameters of current step follow -
     1 - f,x(i)')
  324 format(10d13.5)
  326 format(//'estimated variance matrix')
  329 format(//'estimated standard deviations'/(10d13.5))
  435 format (d19.10,8d13.5/(19x8d13.5))
  436 format(19x8d13.5)
  750 format (/'new absolute minimum , parameters for fcn entry ',i5,
     1'  follow - - f,x(i)')
  785 format (/'step to unallowed region , parameters for fcn entry ',
     1i5,' follow - - x(i)')
  799 format (//'ten consecutive steps to unallowed region - - run termi
     1nated')
  840 format (//'all numbers of last entry refer to final pt. of run')
  865 format (//'the lowest value of the function (fcn) was ',d18.10,
     1' for entry',i5,' - - x(i) follows' )
  875 format(///'run statistics follow - -'//6h nfcn=i5,5h nst=i5)
  911 format(//'there have not been enough steps in the minimisation to
     1make error estimates')
  912 format(//'error analysis assuming chi-square')
  913 format(//39herror analysis assuming -log likelihood)
```

```
 1830 format (//'the function (fcn) has not improved by ',d14.5,' after
     13 steps which include a minimum - - -'///'therefore the run has be
     2en terminated')
 2000 format(3(/),'the maximum number ',i5,' of steps has been reached',
     +       ': run terminated')
      call cputime(1)
c**********************************************
c Set noprint=1 to desactivate the output print.
c Set noprint=0 to activate the output print.
      noprint=0
c**********************************************
      step=1.d0
      al=3.d0
      be=-.4d0
      gam=-2.d0/(al*be)
      neq=0
      nor=0
      amin=1.d308
      call fcn(npar,g,amin,x,1)
      depsi=dble(epsi)
      dwght=dble(wght)
      isw(5)=nlist
c****************************************************************
c JUNE  1993: The following portion of code is used to check the
c             analytical derivatives against the numerical ones, when
c             the log-likelihood function and its analytical
c             derivatives are programmed for the first time.
c             This portion should be desactivated when the analytical
c             derivatives have been checked in order to proceed
c             with the model estimation.
c****************************************************************
c     print*,'log-likelihood = ',amin
c     print*,'no.    parameter      der.anal.      der.num.'
c     do 10 i=1,npar
c     dx=dabs(x(i)/1.d5)
c     if(x(i).eq.0.d0)dx=1.d-5
c     do 11 j=1,npar
c  11 y(j)=x(j)
c     do 12 j=1,2
c     y(i)=x(i)+dx*((-1.)**j)
c  12 call fcn(npar,g,e(j),y,4)
c     dernum=(e(2)-e(1))/(2*dx)
c  10 print 13,i,x(i),g(i),dernum
c  13 format(i3,2x,3d15.7)
c     stop 7777
c************************************************************
      do 9100 i=1,npar
      dirin(i)=0.d0
 9100 wt(i)=dwght
      if(noprint.eq.0)then
         write(6,105)
         write(6,110)
         write(6,120)npar,nstep,sngl(step),epsi
         write(6,130)(sngl(x(i)),i=1,npar)
         write(6,130)(sngl(wt(i)),i=1,npar)
         write(6,130)(sngl(dirin(i)),i=1,npar)
      endif
```

```
      nfcn=1
      f=amin
      a=amin
      nst=0
      avr=amin
      if(noprint.eq.0)then
         write(6,280)
         write(6,435)amin,(x(i),i=1,npar)
      endif
      df=vp(g,g,npar)
      if(df)88,89,88
   88 dercal=.true.
      sp=step/df
      do 1039 i=1,npar
      do 1031 j=1,npar
 1031 et(i,j)=0.d0
 1039 et(i,i)=dabs(wt(i))
 1006 ast=amin
      do 1008 i=1,npar
      d(i)=g(i)
      dirin(i)=0.d0
      do 1008 j=1,npar
 1008 dirin(i)=dirin(i)-et(i,j)*g(j)
      df=vp(dirin,g,npar)
      if(df)1003,43,1003
 1003 continue
      ss=sp
      if(ss)1048,1049,1048
 1049 ss=-step/df
 1048 continue
      sp=0.d0
      do 1007 j=1,4
 1010 if(dabs(ss*df/amin)-1.d-10)1012,1074,1074
 1012 if(d(1).ne.g(1))go to 5400
 1074 st=ss
      do 1004 i=1,npar
 1004 y(i)=x(i)+st*dirin(i)
      fa=f
      call fcn(npar,p,f,y,2)
      nfcn=nfcn+1
      if(nfcn.ge.nstep) then
         if(noprint.eq.0)write(6,2000) nstep
         go to 810
      endif
      if(dabs(f/fa-2.d0)-epsilon)1081,1080,1080
 1081 nor=nor+1
      ss=.5d0*st
      if(isw(5))1085,1086,1085
 1085 if(noprint.eq.0)then
         write(6,785) nfcn
         write(6,436) (y(i),i=1,npar)
      endif
c 1086 if(nor-10)1007,1007,89
 1086 if(nor-3)1007,1007,89

 1080 nor=0
      dg=vp(dirin,p,npar)
```

```
      if(dabs(dg/df)-1.d10)1078,1079,1079
 1079 ss=.2d0*st
      go to 1010
 1078 da=f-amin
      db=da/st
      b=(dg+df+df-3.d0*db)/st
      c=3.d0*(df+dg-db-db)/st**2
      if(c)900,901,900
  901 if(b)1057,1057,902
  902 ss=df*.5d0/b
      go to 903
  900 dfg=b*b-df*c
      if(dfg)1057,1070,1070
 1057 if(da)1059,1056,1056
 1056 ss=-st
      go to 1007
 1070 ss=(b+dsqrt(dfg))/c
  903 if(da)1009,1007,1007
 1059 ss=st*4.d0
 1009 amin=f
      npfn=nfcn
      do 1011 i=1,npar
      g(i)=p(i)
 1011 x(i)=y(i)
      df=dg
      sp=sp+st
      ss=ss-st
      go to 1010
 1007 continue
      go to 1005
 5400 continue
      do 1013 i=1,npar
 1013 d(i)=g(i)-d(i)
      da=0.d0
      db=0.d0
      do 1014 i=1,npar
      y(i)=0.d0
      do 1015 j=1,npar
 1015 y(i)=y(i)+et(i,j)*d(j)
      da=da+dirin(i)*d(i)
 1014 db=db+d(i)*y(i)
      da=sp/da
      do 1016 i=1,npar
      do 1016 j=i,npar
      et(i,j)=et(i,j)+dirin(i)*dirin(j)*da-y(i)*y(j)/db
 1016 et(j,i)=et(i,j)
 1005 nst=nst+1
      if (isw(5))1742,1755,1742
 1742 if(avr-amin)1755,1755,1744
 1744 if(noprint.eq.0)then
         write(6,750) npfn
         write(6,435)amin,(x(i),i=1,npar)
      endif
      avr=amin
 1755 continue
      if (isw(1))810,1808,810
 1808 if (nst-nstep)1035,810,810
```

```
1035 continue
     ame=amin-ast
     go to 40
  89 dercal=.false.
     np=0
     do 39 i=1,npar
     if(wt(i))30,39,30
  30 np=np+1
     do 31 j=1,npar
  31 xs(np,j)=0.d0
     s(np)=0.d0
     xs(np,i)=1.d0
     if(dirin(i))33,34,33
  34 dirin(i)=wt(i)
  33 dirin(np)=dirin(i)*step
  39 continue
     np1=np+1
     s(np1)=0.d0
  15 continue
     do 53 i=1,npar
  53 xs(np1,i)=xs(1,i)
     dirin(np1)=dirin(1)
     am=0.d0
     jm=0
     do 35 j=1,np1
     y1=1.d0
     ne=0
     ns=0
     d(j)=0.d0
   6 do 1 i=1,npar
   1 y(i)=x(i)+dirin(j)*xs(j,i)
     sa=a
     call fcn(npar,g,a,y,4)
     nfcn=nfcn+1
     if(nfcn.ge.nstep) then
        if(noprint.eq.0)write(6,2000) nstep
        go to 810
     endif
     if(dabs(a/sa-2.d0)-epsilon)99,98,98
  99 nor=nor+1
     if(isw(5))97,96,97
  97 if(noprint.eq.0)then
        write(6,785) nfcn
        write(6,436) (y(i),i=1,npar)
     endif
  96 if(nor-10)3,3,95
  95 if(noprint.eq.0)write(6,799)
     go to 810
  98 nor=0
     y2=amin-a
     if(y2)3,18,2
  18 if(y1)38,25,38
   2 ne=0
     amin=a
  38 y1=y2
     npfn=nfcn
     do 4 i=1,npar
```

```
    4 x(i)=y(i)
      d(j)=d(j)+dirin(j)
      dirin(j)=al*dirin(j)
      ns=1
      go to 6
    3 dirin(j)=be*dirin(j)
      if(ns)5,6,5
    5 if(nfcn)25,45,45
   45 if(y1)24,36,24
   36 if(ne)37,37,25
   37 dirin(j)=gam*dirin(j)
      ne=1
      go to 6
   24 x2=dirin(j)/be
      x1=x2/al
      da=x2*y1
      db=x1*y2
      stj=.5d0*(da*x2+db*x1)/(da-db)
      if(stj)27,25,27
   27 do 26 i=1,npar
   26 y(i)=x(i)+stj*xs(j,i)
      sa=a
      call fcn(npar,g,a,y,4)
      nfcn=nfcn+1
      if(nfcn.ge.nstep) then
         if(noprint.eq.0)write(6,2000) nstep
         go to 810
      end if
      if(dabs(a/sa-2.d0)-epsilon)94,93,93
   94 if(isw(5))92,25,92
   92 if(noprint.eq.0)then
        write(6,785) nfcn
        write(6,436) (y(i),i=1,npar)
      endif
      go to 25
   93 if(a-amin)28,25,25
   28 amin=a
      npfn=nfcn
      do 32 i=1,npar
   32 x(i)=y(i)
      d(j)=d(j)+stj
   25 continue
      nst=nst+1
      av=avr-amin
      avr=amin
      if(j-1)741,741,748
  741 ast=amin
      go to 743
  748 if(av-am)743,743,747
  747 am=av
      jm=j
  743 if (isw(5)) 742,755,742
  742 if(av)755,755,744
  744 if(noprint.eq.0)then
         write(6,750) npfn
         write(6,435)amin,(x(i),i=1,npar)
      endif
```

```
755 continue
    if (isw(1)) 810,808,810
808 if (nst-nstep) 35 ,810,810
 35 continue
    ame=amin-ast
    if(ame)40,43,43
 43 if(noprint.eq.0)then
       write(6,44)
       write(6,435)amin,(x(i),i=1,npar)
    endif
    go to 810
 40 if(depsi+ame)20,41,41
 41 neq=neq+1
    if(neq-3)86,21,21
 86 if(dercal) go to 1006
    go to 42
 21 continue
    if(noprint.eq.0)then
       write(6,1830) epsi
       write(6,435)amin,(x(i),i=1,npar)
    endif
810 continue
    if (isw(5)) 838,846,838
838 if(noprint.eq.0)write(6,840)
846 if (isw(4)) 1850,850,1850
 20 neq=0
    if(dercal) go to 1006
 42 continue
    do 7 k=1,npar
    p(k)=0.d0
    do 7 i=2,np1
  7 p(k)=p(k)+d(i)*xs(i,k)
    dp=1.d0
 69 do 63 i=1,npar
 63 y(i)=x(i)+p(i)
    sa=a
    call fcn(npar,g,a,y,4)
    nfcn=nfcn+1
    if(nfcn.ge.nstep) then
       if(noprint.eq.0)write(6,2000) nstep
       go to 810
    end if
    if(dabs(a/sa-2.d0)-epsilon)15,15,65
 65 if(a-amin)66,67,67
 66 amin=a
    dp=dp+dp
    do 68 i=1,npar
    x(i)=y(i)
 68 p(i)=2.d0*p(i)
    go to 69
 67 dp=.5d0*(a+ast-amin-amin)/dp**2
    if(dp-am)64,64,15
 64 if(jm-np)61,61,62
 61 do 8 i=jm,np
    s(i)=s(i+1)
    dirin(i)=dirin(i+1)
    do 8 j=1,npar
```

```
    8 xs(i,j)=xs(i+1,j)
   62 xn=0.d0
      do 14 k=1,npar
      xs(1,k)=p(k)
   14 xn=xn+xs(1,k)**2
      dir=dsqrt(xn)
      dirin(1)=dir
      s(np1)=dp*2.d0
      do 9 k=1,npar
    9 xs(1,k)=xs(1,k)/dir
      go to 15
  910 if(noprint.eq.0)write(6,911)
      go to  850
 1850 if(nst.lt.npar) go to 910
      if(nst.lt.npar**2.and..not.dercal) go to 910
      if(dercal) go to 87
      s(1)=s(np1)
      f=amin
      do 302 i=1,np
      if(s(i))303,303,302
  303 s(i)=1.d+4
  302 p(i)=1.d0/dsqrt(s(i))
      do 312 l=1,2
  311 do 301 i=1,np
  309 do 308 k=1,2
      do 304 j=1,npar
  304 y(j)=x(j)+p(i)*xs(i,j)
      fa=f
      call fcn(npar,g,f,y,4)
      if(dabs(f/fa-2.d0)-epsilon)305,306,306
  306 if(f-amin)307,307,328
  307 do 310 j=1,npar
  310 x(j)=y(j)
      p(i)=2.d0*p(i)
      amin=f
      go to 311
  328 p(i)=-p(i)
  308 e(k)=f
      s(i)=(e(1)+e(2)-amin-amin)/p(i)**2
      if(s(i))305,305,313
  305 p(i)=.5d0*p(i)
      go to 309
  313 s(i)=1.d0/s(i)
  301 p(i)=dsqrt(s(i))
  312 continue
      do 316 l=1,npar
      do 316 m=l,npar
      et(l,m)=0.d0
      do 317 i=1,np
  317 et(l,m)=et(l,m)+xs(i,l)*xs(i,m)*s(i)
  316 et(m,l)=et(l,m)
   87 if(isw(6)) 1050,951,1050
  951 if(noprint.eq.0)write(6,913)
      go to 1051
 1050 if(noprint.eq.0)write(6,912)
      do 1022 i=1,npar
      do 1022 j=1,npar
```

```
 1022 et(i,j)=2.d0*et(i,j)
 1051 do 1023 i=1,npar
      if(et(i,i))1024,1024,1025
 1024 p(i)=0.d0
      go to 1023
 1025 p(i)=dsqrt(et(i,i))
 1023 continue
      if(noprint.eq.0)then
         write(6,326)
         do 327 i=1,npar
  327    write(6,324) (et(i,j),j=1,npar)
         write(6,329)(p(i),i=1,npar)
      endif
  850 continue
      if(noprint.eq.0)then
         write(6,865)amin,npfn
         write(6,130)(sngl(x(i)),i=1,npar)
         write(6,875)nfcn,nst
      endif
      call fcn(npar,g,amin,x,3)
      call cputime(2)
      stop
      end
c
c     ...................................................................
c
c        subroutine loc
c
c        purpose
c           compute a vector subscript for an element in a matrix of
c           specified storage mode
c
c        usage
c           call loc (i,j,ir,n,m,ms)
c
c        description of parameters
c           i    - row number of element
c           j    - column number  of element
c           ir   - resultant vector subscript
c           n    - number of rows in matrix
c           m    - number of columns in matrix
c           ms   - one digit number for storage mode of matrix
c                  0 - general
c                  1 - symmetric
c                  2 - diagonal
c
c        remarks
c           none
c
c        subroutines and function subprograms required
c           none
c
c        method
c           ms=0   subscript is computed for a matrix with n*m elements
c                  in storage (general matrix)
c           ms=1   subscript is computed for a matrix with n*(n+1)/2 in
c                  storage (upper triangle of symmetric matrix). if
```

```
c                  element is in lower triangular portion, subscript is
c                  corresponding element in upper triangle.
c           ms=2   subscript is computed for a matrix with n elements
c                  in storage (diagonal elements of diagonal matrix).
c                  if element is not on diagonal (and therefore not in
c                  storage), ir is set to zero.
c
c     ..................................................................
c
      subroutine loc(i,j,ir,n,m,ms)
c
      ix=i
      jx=j
      if(ms-1) 10,20,30
   10 irx=n*(jx-1)+ix
      go to 36
   20 if(ix-jx) 22,24,24
   22 irx=ix+(jx*jx-jx)/2
      go to 36
   24 irx=jx+(ix*ix-ix)/2
      go to 36
   30 irx=0
      if(ix-jx) 36,32,36
   32 irx=ix
   36 ir=irx
      return
      end

c
c     ..................................................................
c
c        subroutine mata
c
c        purpose
c           premultiply a matrix by its transpose to form a
c           symmetric matrix
c
c        usage
c           call mata(a,r,n,m,ms)
c
c        description of parameters
c           a  - name of input matrix
c           r  - name of output matrix
c           n  - number of rows in a
c           m  - number of columns in a. also number of rows and
c                number of columns of r.
c           ms  - one digit number for storage mode of matrix a
c                  0 - general
c                  1 - symmetric
c                  2 - diagonal
c
c        remarks
c           matrix r cannot be in the same location as matrix a
c           matrix r is always a symmetric matrix with a storage mode=1
c
c        subroutines and function subprograms required
c           loc
```

```
c
c        method
c           calculation of (a transpose a) results in a symmetric matrix
c           regardless of the storage mode of the input matrix. the
c           elements of matrix a are not changed.
c
c     ..................................................................
c
      subroutine mata(a,r,n,m,ms)
      implicit double precision (a-h,o-z)
      dimension a(1),r(1)
c
      do 60 k=1,m
      kx=(k*k-k)/2
      do 60 j=1,m
      if(j-k) 10,10,60
   10 ir=j+kx
      r(ir)=0.
      do 160 i=1,n
      if(ms) 20,40,20
   20 call loc(i,j,ia,n,m,ms)
      call loc(i,k,ib,n,m,ms)
      if(ia) 30,160,30
   30 if(ib) 50,160,50
   40 ia=n*(j-1)+i
      ib=n*(k-1)+i
   50 r(ir)=r(ir)+a(ia)*a(ib)
  160 continue
   60 continue
      return
      end

      subroutine mcpy(a,r,n,m,ms)
c Copy matrix A in matrix R:
c   A  - Name of input matrix.
c   R  - Name of output matrix.
c   N  - Number of rows in A or R.
c   M  - Number of columns in A or R.
c   MS - One digit number for storage mode of matrix A (and R):
c        0 - General.
c        1 - Symmetric.
c        2 - Diagonal.
c Subroutines and function subprograms required:
c   LOC
      double precision a(*),r(*)
      call loc(n,m,it,n,m,ms)
      do 1 i=1,it
1     r(i)=a(i)
      return
      end

c
c     ..................................................................
c
c        subroutine mfsd
c
c        purpose
```

```
c           factor a given symmetric positive definite matrix
c
c        usage
c           call mfsd(a,n,eps,ier)
c
c        description of parameters
c           a      - upper triangular part of the given symmetric
c                    positive definite n by n coefficient matrix.
c                    on return a contains the resultant upper
c                    triangular matrix.
c           n      - the number of rows (columns) in given matrix.
c           eps    - an input constant which is used as relative
c                    tolerance for test on loss of significance.
c           ier    - resulting error parameter coded as follows
c                    ier=0  - no error
c                    ier=-1 - no result because of wrong input parame-
c                             ter n or because some radicand is non-
c                             positive (matrix a is not positive
c                             definite, possibly due to loss of signi-
c                             ficance)
c                    ier=k  - warning which indicates loss of signifi-
c                             cance. the radicand formed at factoriza-
c                             tion step k+1 was still positive but no
c                             longer greater than abs(eps*a(k+1,k+1)).
c
c        remarks
c           the upper triangular part of given matrix is assumed to be
c           stored columnwise in n*(n+1)/2 successive storage locations.
c           in the same storage locations the resulting upper triangu-
c           lar matrix is stored columnwise too.
c           the procedure gives results if n is greater than 0 and all
c           calculated radicands are positive.
c           the product of returned diagonal terms is equal to the
c           square-root of the determinant of the given matrix.
c
c        subroutines and function subprograms required
c           none
c
c        method
c           solution is done using the square-root method of cholesky.
c           the given matrix is represented as product of two triangular
c           matrices, where the left hand factor is the transpose of
c           the returned right hand factor.
c
c     ..................................................................
c
      subroutine mfsd(a,n,eps,ier)
c
c
      implicit double precision (a-h,o-z)
      dimension a(1)
c
c        test on wrong input parameter n
      if(n-1) 12,1,1
    1 ier=0
c
c        initialize diagonal-loop
```

```
      kpiv=0
      do 11 k=1,n
      kpiv=kpiv+k
      ind=kpiv
      lend=k-1
c
c        calculate tolerance
      tol=dabs(eps*a(kpiv))
c
c        start factorization-loop over k-th row
      do 11 i=k,n
      dsum=0.d0
      if(lend) 2,4,2
c
c        start inner loop
    2 do 3 l=1,lend
      lanf=kpiv-l
      lind=ind-l
    3 dsum=dsum+a(lanf)*a(lind)
c        end of inner loop
c
c        transform element a(ind)
    4 dsum=a(ind)-dsum
      if(i-k) 10,5,10
c
c        test for negative pivot element and for loss of significance
    5 if(dsum-tol) 6,6,9
    6 if(dsum) 12,12,7
    7 if(ier) 8,8,9
    8 ier=k-1
c
c        compute pivot element
    9 dpiv=dsqrt(dsum)
      a(kpiv)=dpiv
      dpiv=1.d0/dpiv
      go to 11
c
c        calculate terms in row
   10 a(ind)=dsum*dpiv
   11 ind=ind+i
c
c        end of diagonal-loop
      return
   12 ier=-1
      return
      end

      subroutine minv(a,n,d,l,m,ier)
      double precision a,d,epsdet,biga,hold
      dimension a(*),l(*),m(*)
      data epsdet/1.d-300/
c******************************************************************************
c Invert a general matrix:
c a - Input matrix, destroyed in computation and replaced by resultant inverse.
c n - Order of matrix a.
c d - Resultant determinant of matrix a.
c l - Work vector of length n.
```

```
c m - Work vector of length n.
c ier    - error code for matrix singularity:
c          ier = 0 (no singularity) and 1 (matrix singularity).
c epsdet - Input constant such that a determinant with absolute value less than
c          epsdet indicates that matrix a is singular (ier=1).

c Method:
c     The standard Gauss-Jordan method is used.
c     The determinant is also calculated.
c*******************************************************************************
c Search for largest element
      ier=0
      d=1.d0
      nk=-n
      do 80 k=1,n
      nk=nk+n
      l(k)=k
      m(k)=k
      kk=nk+k
      biga=a(kk)
      do 20 j=k,n
      iz=n*(j-1)
      do 20 i=k,n
      ij=iz+i
10    if(dabs(biga)-dabs(a(ij)))15,20,20
15    biga=a(ij)
      l(k)=i
      m(k)=j
20    continue
c Interchange rows
      j=l(k)
      if(j-k)35,35,25
25    ki=k-n
      do 30 i=1,n
      ki=ki+n
      hold=-a(ki)
      ji=ki-k+j
      a(ki)=a(ji)
30    a(ji)=hold
c Interchange columns
35    i=m(k)
      if(i-k)45,45,38
38    jp=n*(i-1)
      do 40 j=1,n
      jk=nk+j
      ji=jp+j
      hold=-a(jk)
      a(jk)=a(ji)
40    a(ji)=hold
c Divide column by minus pivot (value of pivot element is contained in biga)
45    if(dabs(biga)-epsdet)46,46,48
c Matrix singularity
46    ier=1
      d=0.d0
      return
48    do 55 i=1,n
      if(i-k)50,55,50
```

```
50    ik=nk+i
      a(ik)=a(ik)/(-biga)
55    continue
c Reduce matrix
      do 65 i=1,n
      ik=nk+i
      hold=a(ik)
      ij=i-n
      do 65 j=1,n
      ij=ij+n
      if(i-k)60,65,60
60    if(j-k)62,65,62
62    kj=ij-i+k
      a(ij)=hold*a(kj)+a(ij)
65    continue
c Divide row by pivot
      kj=k-n
      do 75 j=1,n
      kj=kj+n
      if(j-k)70,75,70
70    a(kj)=a(kj)/biga
75    continue
c Product of pivots
      d=d*biga
c Replace pivot by reciprocal
      a(kk)=1.d0/biga
80    continue
c Final row and column interchange
      k=n
100   k=k-1
      if(k)150,150,105
105   i=l(k)
      if(i-k)120,120,108
108   jq=n*(k-1)
      jr=n*(i-1)
      do 110 j=1,n
      jk=jq+j
      hold=a(jk)
      ji=jr+j
      a(jk)=-a(ji)
110   a(ji)=hold
120   j=m(k)
      if(j-k)100,100,125
125   ki=k-n
      do 130 i=1,n
      ki=ki+n
      hold=a(ki)
      ji=ki-k+j
      a(ki)=-a(ji)
130   a(ji)=hold
      go to 100
150   return
      end

c
c     .................................................................
c
```

```
c        subroutine mprd
c
c        purpose
c           multiply two matrices to form a resultant matrix
c
c        usage
c           call mprd(a,b,r,n,m,msa,msb,l)
c
c        description of parameters
c           a - name of first input matrix
c           b - name of second input matrix
c           r - name of output matrix
c           n - number of rows in a and r
c           m - number of columns in a and rows in b
c           msa - one digit number for storage mode of matrix a
c                  0 - general
c                  1 - symmetric
c                  2 - diagonal
c           msb - same as msa except for matrix b
c           l - number of columns in b and r
c
c        remarks
c           matrix r cannot be in the same location as matrices a or b
c           number of columns of matrix a must be equal to number of row
c           of matrix b
c
c        subroutines and function subprograms required
c           loc
c
c        method
c           the m by l matrix b is premultiplied by the n by m matrix a
c           and the result is stored in the n by l matrix r. this is a
c           row into column product.
c           the following table shows the storage mode of the output
c           matrix for all combinations of input matrices
c                        a                b                r
c                     general          general          general
c                     general          symmetric        general
c                     general          diagonal         general
c                     symmetric        general          general
c                     symmetric        symmetric        general
c                     symmetric        diagonal         general
c                     diagonal         general          general
c                     diagonal         symmetric        general
c                     diagonal         diagonal         diagonal
c
c     ..................................................................
c
      subroutine mprd(a,b,r,n,m,msa,msb,l)
      implicit double precision (a-h,o-z)
      dimension a(1),b(1),r(1)
c
c        special case for diagonal by diagonal
c
      ms=msa*10+msb
      if(ms-22) 30,10,30
   10 do 20 i=1,n
```

```
   20 r(i)=a(i)*b(i)
      return
c
c        all other cases
c
   30 ir=1
      do 90 k=1,l
      do 90 j=1,n
      r(ir)=0.
      do 80 i=1,m
      if(ms) 40,60,40
   40 call loc(j,i,ia,n,m,msa)
      call loc(i,k,ib,m,l,msb)
      if(ia) 50,80,50
   50 if(ib) 70,80,70
   60 ia=n*(i-1)+j
      ib=m*(k-1)+i
   70 r(ir)=r(ir)+a(ia)*b(ib)
   80 continue
   90 ir=ir+1
      return
      end

c
c     ..................................................................
c
c        subroutine ndtr
c
c        purpose
c           computes y = p(x) = probability that the random variable  u,
c           distributed normally(0,1), is less than or equal to x.
c           f(x), the ordinate of the normal density at x, is also
c           computed.
c
c        usage
c           call ndtr(x,p,d)
c
c        description of parameters
c           x--input scalar for which p(x) is computed.
c           p--output probability.
c           d--output density.
c
c        remarks
c           maximum error is 0.0000007.
c
c        subroutines and subprograms required
c           none
c
c        method
c           based on approximations in c. hastings, approximations for
c           digital computers, princeton univ. press, princeton, n.j.,
c           1955.  see equation 26.2.17, handbook of mathematical
c           functions, abramowitz and stegun, dover publications, inc.,
c           new york.
c
c     ..................................................................
c
```

```
      subroutine ndtr(x,p,d)
c
      implicit double precision (a-h,o-z)
      ax=dabs(x)
      t=1.0/(1.0+.2316419*ax)
c Underflow for the exponential of y = -(x*x)/2 for SUN4
      d=0.
      y=-(x*x)/2
      if(y.ge.dlog(1.d-307))d=0.3989423*dexp(y)
      p = 1.0 - d*t*((((1.330274*t - 1.821256)*t + 1.781478)*t -
     1  0.3565638)*t + 0.3193815)
      if(x)1,2,2
    1 p=1.0-p
    2 return
      end

      subroutine pmatrix(rho,rstar1,rstar2,p,n,norder)
c*******************************************************************
c Compute an (N x N) matrix P = I - RHO(1)*RSTAR1, if norder = 1,
c        or   P = I - RHO(1)*RSTAR1 - RHO(2)*RSTAR2, if norder = 2,
c where I is an identity matrix of order N,
c       RHO is the vector containing RHO(1) and RHO(2),
c       RSTAR1 and RSTAR2 are (N x N) general matrices.
c*******************************************************************
      implicit double precision (a-h,o-z)
      dimension rho(*),rstar1(*),rstar2(*),p(*)
      do 1 i=1,n
      do 2 j=1,n
      ij=i+(j-1)*n
2     p(ij)=-rho(1)*rstar1(ij)
      id=i+(i-1)*n
      p(id)=1.d0 + p(id)
1     continue
      if(norder.eq.2)then
         do 3 i=1,n
         do 3 j=1,n
         ij=i+(j-1)*n
3        p(ij)=p(ij) - rho(2)*rstar2(ij)
      endif
      return
      end

      subroutine shminv(r,c,oldinv,binv,n)
c Compute the INVERSE of B = [ INVERSE(R) + cI ] with the SHERMAN-MORRISON
c formula modified to take into account the sparseness of the matrix R.
c INPUT  : R      - square matrix stored columnwise as a vector of length n*n.
c          c      - scalar.
c          I      - identity matrix of order n, which need not to be stored.
c          OLDINV - working square matrix stored columnwise as a vector of
c                   length n*n, containing the inverse of B associated with each
c                   successive change of the k-th diagonal element (k=1,n-1).
c          n      - order of matrices R, OLDINV and BINV.
c OUTPUT : BINV   - square matrix stored columnwise as a vector of length n*n,
c                   containing the inverse of B following a change of the last
c                   diagonal element (k=n).
      implicit double precision (a-h,o-z)
      dimension r(*),oldinv(*),binv(*)
```

```
      call mcpy(r,binv,n,n,0)
      if(dabs(c).le.1.d-15)return
      do 1 k=1,n
      call mcpy(binv,oldinv,n,n,0)
      kk=k+(k-1)*n
      factor=c/(1.+c*oldinv(kk))
      do 1 i=1,n
      ik=i+(k-1)*n
      if(oldinv(ik).ne.0.d0)then
         fold=factor*oldinv(ik)
         do 2 j=1,n
         kj=k+(j-1)*n
         if(oldinv(kj).ne.0.d0)then
            ij=i+(j-1)*n
            binv(ij)=oldinv(ij)-fold*oldinv(kj)
         endif
2        continue
      endif
1     continue
      return
      end

c
c     ..................................................................
c
c        subroutine sinv
c
c        purpose
c           invert a given symmetric positive definite matrix
c
c        usage
c           call sinv(a,n,eps,ier)
c
c        description of parameters
c           a      - upper triangular part of the given symmetric
c                    positive definite n by n coefficient matrix.
c                    on return a contains the resultant upper
c                    triangular matrix.
c           n      - the number of rows (columns) in given matrix.
c           eps    - an input constant which is used as relative
c                    tolerance for test on loss of significance.
c           ier    - resulting error parameter coded as follows
c                    ier=0  - no error
c                    ier=-1 - no result because of wrong input parame-
c                             ter n or because some radicand is non-
c                             positive (matrix a is not positive
c                             definite, possibly due to loss of signi-
c                             ficance)
c                    ier=k  - warning which indicates loss of signifi-
c                             cance. the radicand formed at factoriza-
c                             tion step k+1 was still positive but no
c                             longer greater than abs(eps*a(k+1,k+1)).
c
c        remarks
c           the upper triangular part of given matrix is assumed to be
c           stored columnwise in n*(n+1)/2 successive storage locations.
c           in the same storage locations the resulting upper triangu-
```

```
c           lar matrix is stored columnwise too.
c           the procedure gives results if n is greater than 0 and all
c           calculated radicands are positive.
c
c        subroutines and function subprograms required
c           mfsd
c
c        method
c           solution is done using the factorization by subroutine mfsd.
c
c     ..................................................................
c
      subroutine sinv(a,n,eps,ier)
      implicit double precision (a-h,o-z)
      dimension a(1)
c
c        factorize given matrix by means of subroutine mfsd
c        a = transpose(t) * t
      call mfsd(a,n,eps,ier)
      if(ier) 9,1,1
c
c        invert upper triangular matrix t
c        prepare inversion-loop
    1 ipiv=n*(n+1)/2
      ind=ipiv
c
c        initialize inversion-loop
      do 6 i=1,n
      din=1.d0/a(ipiv)
      a(ipiv)=din
      min=n
      kend=i-1
      lanf=n-kend
      if(kend) 5,5,2
    2 j=ind
c
c        initialize row-loop
      do 4 k=1,kend
      work=0.d0
      min=min-1
      lhor=ipiv
      lver=j
c
c        start inner loop
      do 3 l=lanf,min
      lver=lver+1
      lhor=lhor+l
    3 work=work+a(lver)*a(lhor)
c        end of inner loop
c
      a(j)=-work*din
    4 j=j-min
c        end of row-loop
c
    5 ipiv=ipiv-min
    6 ind=ind-1
c        end of inversion-loop
```

```
c
c        calculate inverse(a) by means of inverse(t)
c        inverse(a) = inverse(t) * transpose(inverse(t))
c        initialize multiplication-loop
      do 8 i=1,n
      ipiv=ipiv+i
      j=ipiv
c
c        initialize row-loop
      do 8 k=i,n
      work=0.d0
      lhor=j
c
c        start inner loop
      do 7 l=k,n
      lver=lhor+k-i
      work=work+a(lhor)*a(lver)
    7 lhor=lhor+l
c        end of inner loop
c
      a(j)=work
    8 j=j+k
c        end of row- and multiplication-loop
c
    9 return
      end

      subroutine smpelast(ymean,xmean,zmean,beta,delta,vly,vlx,vlz,
     +nx,nobs,nullx,nb,nvaz,nlamy,numlx,numlz,numxeqz,namx,
     +smpderx,smpelax,smpderz,smpelaz)
c Input arguments : (ymean,...,namx) (see Definitions in Subroutine FCN).
c Output arguments: (smpderx,...,smpelaz) (see Definitions below).
      implicit double precision (a-h,o-z)
      character*10 namx(nx)

      real*4 epsi,wght

      dimension xmean(*),zmean(*),beta(*),delta(*)
      dimension vlx(*),vlz(*),nullx(*)
      dimension numlx(*),numlz(*),numxeqz(*)
      dimension smpderx(*),smpelax(*),smpderz(*),smpelaz(*)

      common nlist,nstep,epsi,wght,noprint


c Compute at the sample means of Y and X:
c     SMPDERX : derivative of Y (sample value) with respect to X
c and SMPELAX : elasticity of Y (sample value) with respect to X.
      denom=1.0d0
      if(nlamy.eq.1)denom=ymean**(vly-1)
      do 1 k=1,nb
      smpderx(k)=beta(k)
      ngroup=numlx(k)
      if(ngroup.gt.0)smpderx(k)=smpderx(k)*xmean(k)**(vlx(ngroup)-1)
      smpderx(k)=smpderx(k)/denom
      smpelax(k)=smpderx(k)*xmean(k)/ymean
c Correction of the elasticity of Y with respect to a dummy or quasi-dummy
```

```
c for the portion of positive observations of the variable.
      if(nullx(k).gt.0)smpelax(k)=nobs*smpelax(k)/(nobs-nullx(k))
1     continue

c Heteroskedasticity case.
      if(nvaz.gt.0)then
c Compute residual U : Y(BOX-COX) - X(BOX-COX)*BETA evaluated at the sample
c means of Y and X.
         u=ymean
         if(nlamy.eq.1)u=zlam(ymean,vly)
         do 2 k=1,nb
         ngroup=numlx(k)
         if(ngroup.eq.0)then
            u=u-xmean(k)*beta(k)
         else
            u=u-zlam(xmean(k),vlx(ngroup))*beta(k)
         endif
2        continue
c Compute at the sample means of Y, X and Z:
c     SMPDERZ : derivative of Y (sample value) with respect to Z
c and SMPELAZ : elasticity of Y (sample value) with respect to Z.
         do 3 k=1,nvaz
         smpderz(k)=delta(k)*u/2
         ngroup=numlz(k)
         if(ngroup.gt.0)smpderz(k)=smpderz(k)*zmean(k)**(vlz(ngroup)-1)
         smpderz(k)=smpderz(k)/denom
         smpelaz(k)=smpderz(k)*zmean(k)/ymean
c Case where a heteroskedasticity variable Z also appears as an independent
c variable X.
         ix=numxeqz(k)
         if(ix.gt.0)then
            smpderx(ix)=smpderx(ix)+smpderz(k)
            smpelax(ix)=smpelax(ix)+smpelaz(k)
            smpderz(k)=smpderx(ix)
            smpelaz(k)=smpelax(ix)
         endif
3        continue
      endif

c Sept.99: Compute the MRS between two independent variables
c          using the ratio of the derivatives of the sample value of Y
c          with respect to X(l) and X(k), evaluated at the sample means
c          of Y, X and Z.

      if(noprint.eq.0)then
      print 30
30    format(//'MRS of X(l) with respect to X(k) - at the sample means')
      ncol=10
      npage=nb/ncol
      if(nb.gt.npage*ncol)npage=npage+1
      do 31 j=1,npage
         j2=j*ncol
         j1=j2-ncol+1
         if(nb.lt.j2)j2=nb
         print 32
32       format(131(1h=))
         print 33,(namx(k),k=j1,j2)
```

```
33       format(8x,'X(k)',t12,10(2x,a10))
         print 34
34       format('X(l)')
         print 32
         do 35 kr=1,nb
35       print 36,namx(kr),(smpderx(kc)/smpderx(kr),kc=j1,j2)
36       format(/a10,t12,10d12.4)
31    continue

      endif
      return
      end

      subroutine smpy(a,c,r,n,m,ms)
      implicit double precision (a-h,o-z)
      dimension a(*),r(*)
c
c Purpose:
c  Multiply each element of a matrix by a scalar to form a resultant matrix.
c Parameters:
c  A  - Name of input matrix.
c  C  - Input scalar.
c  R  - Name of output matrix.
c  N  - Number of rows in matrix A and R.
c  M  - Number of columns in matrix A and R.
c  MS - One digit number for storage mode of matrix A and R:
c       0 - General.
c       1 - Symmetric.
c       2 - Diagonal.
c Remarks:
c  Matrix R can be in the same location as matrix A.
c Subroutines and function subprograms required:
c  LOC
c
c Compute vector length IT.
      call loc(n,m,it,n,m,ms)
c Multiply each element of A by scalar C.
      do 1 i=1,it
1     r(i)=c*a(i)
      return
      end

      subroutine stormat(derw,dwmat,n,nobs,np,kpar)
      implicit double precision (a-h,o-z)
      dimension derw(n),dwmat(n,np)
c Store the vector DERW of derivatives of residual W with respect to a KPAR-th
c parameter, which can be a THETA, LAMBDA(Y) or LAMBDA(X), in a (N x NP) matrix
c DWMAT,
c where  N = maximum number of observations used in DIMENSION statement of the
c            calling routine,
c       NP = maximum number of THETA's, LAMBDA(Y) and LAMBDA(X)'s, used in
c            DIMENSION statement of the calling routine: NTHETA + 1 + NLX
c and NOBS = number of observations used in estimation.
c The matrix DWMAT is used to compute the covariance matrix of the estimated
c parameters at convergence of the log-likelihood function.
      do 1 i=1,nobs
1     dwmat(i,kpar)=derw(i)
```

```
      return
      end

      double precision function trace(a,b,n)
c Compute the trace of the product of two general matrices A and B of order N.
      double precision a(*),b(*)
      trace=0.d0
      do 1 i=1,n
      do 1 j=1,n
      ij=i+(j-1)*n
      ji=j+(i-1)*n
1     trace=trace+a(ij)*b(ji)
      return
      end

      subroutine vardec(x,xx,namx,n,nobs,nb,nxx)
c INPUT:
c   X     (n x nb) Matrix of (original or transformed) Independent Variables,
c         including the Constant, used in ESTIMATION.
c   NAMX  (nb) Vector of Names of the Independent Variables, including the
c         Constant, used in ESTIMATION.
c   N     Maximum Number of Observations as declared on the Parameter card
c         in Subroutine FCN.
c   NOBS  Number of Observations used in ESTIMATION.
c   NB    Number of Independent Variables, including the Constant, used in
c         ESTIMATION.
c   NXX   Maximum Number of Different Elements in the (nx x nx) Symmetric
c         Matrix XX, stored columnwise as an upper triangular matrix.
c         This number is declared on the Parameter card in Subroutine FCN.
c WORK VECTOR:
c   XX    (nx x nx) Symmetric Matrix, stored columnwise as an upper triangular
c         matrix.
c OUTPUT:
c   D     (30) Vector of Eigenvalues in increasing order.
c   C     (30) Vector of Condition Indices.
c   Z     (30 x 30 = 900) Matrix of Variance-Decomposition Proportions, stored
c         columnwise.
c PURPOSE:
c   Compute the Table of Variance-Decomposition Proportions for the (original
c   or transformed) Independent Variables X, including the Constant.

      implicit double precision (a-h,o-z)
      character*10 namx(nb)

      real*4 epsi,wght

      dimension x(n,nb),xx(nxx)
      dimension c(30),d(30),z(900)

      common nlist,nstep,epsi,wght,noprint
c Compute Matrix X'X with the Columns of Matrix X not normalized to unit length.
      ir=0
      do 1 j1=1,nb
      do 1 j2=1,j1
      ir=ir+1
    1 xx(ir)=0.
      do 6 i=1,nobs
```

```
      ir=0
      do 6 j1=1,nb
      do 6 j2=1,j1
      ir=ir+1
    6 xx(ir)=xx(ir)+x(i,j1)*x(i,j2)
c Compute Matrix X'X with the Columns of Matrix X normalized to unit length.
      do 7 j=1,nb
      ir=(j*(j+1))/2
    7 d(j)=xx(ir)
      ir=0
      do 8 j1=1,nb
      do 8 j2=1,j1
      ir=ir+1
    8 xx(ir)=xx(ir)/dsqrt(d(j1)*d(j2))
c Compute the Eigenvalues (stored in Vector d) and the Eigenvectors (stored
c temporarily in Matrix z) associated with Matrix X'X, with the Columns
c of Matrix X normalized to unit length.
      call eigen(xx,z,nb,0)
      do 9 j=1,nb
      ir=(j*(j+1))/2
    9 d(j)=xx(ir)
      do 3 k=1,nb
      c(k)=0.
      do 3 j=1,nb
      call loc(k,j,ir,nb,nb,0)
    3 c(k)=c(k)+(z(ir)**2)/d(j)
      do 4 k=1,nb
      do 4 j=1,nb
      call loc(k,j,ir,nb,nb,0)
    4 z(ir)=z(ir)**2/(d(j)*c(k))
      do 2 i=1,nb
    2 c(i)=dsqrt(d(nb)/d(i))



      if(noprint.eq.0)then
      np=nb/10
      if(mod(nb,10).ne.0)np=np+1
      do 300 i=1,np
      j2=i*10
      j1=j2-9
      if(j2.gt.nb)j2=nb
      print 100
  100 format(///'no.  eigenvalue'2x'condition index'2x
     +'variance decomposition proportions'/)
      print 150,(namx(j),j=j1,j2)
  150 format(/34x,10a10)
      do 5 k=1,nb
      call loc(j1,k,jk1,nb,nb,0)
      call loc(j2,k,jk2,nb,nb,0)
    5 print 200,k,d(k),c(k),(z(j),j=jk1,jk2)
  200 format(/i3,f12.4,f15.2,10f10.4)
  300 continue

      endif
```

```
      return
      end

      double precision function vp(a,b,n)
      double precision a,b
      dimension a(*),b(*)
c Vector product of A*B
c where A is a (1 x N) vector and B is an (N x 1) vector.
      vp=0.
      do 1 i=1,n
    1 vp=vp+a(i)*b(i)
      return
      end

      double precision function zlam(z,vl)
c   compute the box-cox transformation z(lambda) where vl is a value of lambda.
      implicit double precision (a-h,o-z)
      zlam=0.
      if(z.eq.0.)return
      if(dabs(vl).ge.1d-4)go to 1
      zlam=dlog(z)
      return
    1 zlam=(z**vl-1)/vl
      return
      end
```

3.2 Program L-2.1 Tablex

The program consists of one main module and two subroutines. It was compiled with Compaq Visual Fortran 6.6.

Program listing

```
      program L2tablex
c*******************************************************************************
c   January 2012: STANDARD TABLEX for LEVEL L-2.1
c
c   UNIT=1 (formatted) READ pour le fichier des definitions des blocs et
c                      des variables X et Z, et des noms et définitions
c                      des matrices de contiguité.
c   UNIT=2 (formatted) READ pour les fichiers de resultats des variantes
c                      estimees par le programme Level1.
c
c   Note importante:   Contrairement a TABLEX de LEVEL1, on ne lit plus dans
c                      TABLEX de LEVEL2 le fichier de donnees contenant toutes
c                      les variables Y et X, dont les variables quasi-dummies
c                      et dummies qui sont necessaires pour la correction des
c                      elasticites de Y par rapport a ces variables.
c                      Cette correction est deja faite dans le programme
c                      LEVEL2 lui-meme, contrairement a LEVEL1.
c*******************************************************************************
      implicit double precision (a-h,o-z)
c Maximum dimensions for the arrays:
c   maxeq   = maximum number of equations (or variants) reported in a Tablex.
c   maxvar  = maximum number of independent variables (including the heteroske-
c             dasticity variables Z).
c   maxlamx = maximum number of Lambda(x)'s.
c   maxzhet = maximum number of variables Z used in heteroskedasticity.
c   maxmat  = maximum number of contiguity matrices R.
      parameter (maxeq=10,maxvar=350,maxlamx=10,maxzhet=5,maxmat=10)
      character*100 fname
      character*35 matdef(maxmat),idef,idefz(maxzhet),stitle(9)
      character*10 fmt(4),ifmt(2),alpha(maxeq),alphax(maxeq,9)
      character*9 varname,nvtname(maxeq)
      character*8 namv(maxvar),namy(maxeq)
      character*8 namz(maxzhet),name,nambl
      character*8 namx,namxref(maxeq),xrefname(maxeq),modtype(maxeq)
      character*8 matbloc,matname(maxmat),imatname(maxmat)

      character*3 nty(maxvar,maxeq),ntypelx
      character*1 icar(35),under(8)

      dimension nvax(maxeq),norder(maxeq),nvaz(maxeq),nlamz(maxeq)
      dimension nlamy(maxeq),nlamx(maxeq),nleq(maxeq)
      dimension deryxref(maxeq),deryx(2),ela(2),coryx(2),coryz(2)
      dimension numv(maxvar),x(maxvar,maxeq,9),ngr(maxvar,maxeq)
      dimension nbeta(maxvar,maxeq),nvi(maxeq),npar(maxeq)
      dimension ivernum(maxeq)
      dimension ntly(maxeq),ntlx(maxlamx,maxeq)
      dimension vly(maxeq,3),vlx(maxlamx,maxeq,3)
```

```
      dimension ntdel(maxzhet,maxeq),ntlz(maxzhet,maxeq)
      dimension vd(maxzhet,maxeq,9),vlz(maxzhet,maxeq,3),numlz(maxzhet)
      dimension smpderz(maxzhet),smpelaz(maxzhet)
      dimension expderz(maxzhet),expelaz(maxzhet)

      dimension ntrho(maxmat,maxeq),ntpi(maxmat,maxeq)
      dimension rho(maxmat,maxeq,2),nr(maxmat),pi(maxmat,maxeq,3)

      dimension r2p(maxeq),r2pa(maxeq),r2(maxeq),r2bar(maxeq)
      dimension vlik(maxeq),plim(maxeq),ymean(maxeq),eymean(maxeq)
      dimension nlyfix(maxeq),nlyestcon(maxeq),nlyestdis(maxeq)
      dimension nlxfix(maxeq),nlxest(maxeq)
      dimension ndelfix(maxeq),ndelest(maxeq)
      dimension nlzfix(maxeq),nlzest(maxeq)
      dimension nroest(maxeq),npifix(maxeq),npiest(maxeq)
      dimension ncst(maxeq)
      dimension nstat(9)

      namelist/card/nbloc,nmatr,neq,ndt,nvs,nlist,nstat
c**************************************************************************
c Input parameters in namelist/card/:
c     nbloc  : Number of blocks of independent variables X in the
c              list of definitions of X (number of blocks unlimited for a total
c              of 350 independent variables).
c     nmatr  : Number of contiguity matrices R in the list of definitions
c              of the matrices (max. 10 matrices).
c     neq    : Number of equations (or variants) to be reported
c              in TABLEX (max. 10 equations).
c     ndt    : Number of variables used in the heteroskedasticity
c              function for the "neq" equations (max.5 variables).
c     nvs    : Language version (nvs = 0 - French, nvs = 1 - English)
c     nlist  : 0 - Don't print the list of definitions of the variables.
c              1 - Print the list of definitions of the variables.
c     nstat  : 0 - Don't print the ith statistic for Beta or Delta.
c            : i - Print the ith statistic for Beta or Delta (i=1,...,9).
c**************************************************************************
      data nbeta/3500*0/
      data x/31500*0./
      data vd/450*0/
c All the following variables (nt...) indicate the type of the parameters
c (ntly for lambda(y), ntlx for lambda(x), ntdel for delta,
c  ntlz for lambda(z), ntrho for rho and ntpi for pi).
c If the type is equal to -1, the parameter is not specified in the equation.
c If the type is equal to 0, the parameter is fixed in the equation.
c If the type is equal to 1, the parameter is estimated in the equation.
      data ntly/10*-1/
      data ntlx/100*-1/
      data ntdel/50*-1/
      data ntlz/50*-1/
      data ntrho/100*-1/
      data ntpi/100*-1/

c Lire les parametres de controle du namelist/card/
c Lire le nom de la variable X_ref utilisee comme variable de reference par
c   rapport a laquelle tous les taux maginaux de substitution des autres
c   variables independantes X_k seront calcules, pour chacune des "neq"
variantes.
```

```
c Lire les noms des variables qui sont utilisees pour l'heteroscedasticite
c   dans tout l'ensemble des variantes reportees dans TABLEX, si ndt > 0.

      read card
c xrefname est le nom de X_ref cadré à gauche.
c namxref est le nom de X_ref cadré à droite.
      do 6 i=1,neq
c Initialize the numbers of fixed and estimated parameters for:
c Lambda(y)
      nlyfix(i)=0
      nlyestdis(i)=0
      nlyestcon(i)=0
c Lambda(x)
      nlxfix(i)=0
      nlxest(i)=0
c Delta(z)
      ndelfix(i)=0
      ndelest(i)=0
c Lambda(z)
      nlzfix(i)=0
      nlzest(i)=0
c Rho
      nroest(i)=0
c Pi
      npifix(i)=0
      npiest(i)=0
c Constant
      ncst(i)=1

      read 3,xrefname(i)
      namxref(i)=adjustr(xrefname(i))
    6 continue
      if(ndt.gt.0)then
         do 1000 i=1,ndt
         read 3,namz(i)
    3    format(a8)
 1000    continue
      endif

      nvar=0
      if(nlist.eq.1)print 1901
 1901 format(///)

c Lire le fichier contenant la liste des noms (8 caracteres) et des definitions
c (35 caracteres) des blocs et des variables X sur UNIT=1

      read 29,fname
   29 format(a100)
      open(unit=1,file=fname)
c Il y a 'nbloc' de variables independantes X.
c 'nvar' est le total des variables independantes X lues dans le fichier.
      nvar=0
      do 1 i=1,nbloc
      if(nvs.eq.0)read(1,1001)num,nambl,idef
      if(nvs.eq.1)read(1,1002)num,nambl,idef
 1001 format(i2,a8,a35)
 1002 format(i2,a8,35x,a35)
```

```
      if(nlist.ne.0)print 1003,i,nambl,idef
 1003 format(//i5,5x,a8,5x,a35)
   10 continue
c Pour une variable X, numv est la catégorie de variable: 0,1,2,3.
c     2 : Quasi-dummy
c     3 : Dummy
      if(nvs.eq.0)read(1,1001)num,namx,idef
      if(nvs.eq.1)read(1,1002)num,namx,idef
      if(namx.eq.' ')go to 1
      nvar=nvar+1
      numv(nvar)=num
      namv(nvar)=namx
c Stocker les définitions des variables Z utilisées dans l'hétéroscédasticité.
      do 5 j=1,ndt
      if(namx.eq.namz(j))idefz(j)=idef
    5 continue

c Imprimer la liste des definitions des blocs et des variables.
      if(nlist.ne.0)print 1004,num,idef,namv(nvar)
 1004 format(/10x,i5,5x,a35,5x,a8)
      go to 10
    1 continue

c Lire les noms (8 caracteres) et les definitions (35 caracteres)
c des matrices R. Il y a 'nmatr' matrices R.
      read(1,499)matbloc
  499 format(2x,a8)
      do 4 i=1,nmatr
      if(nvs.eq.0)read(1,500)matname(i),matdef(i)
      if(nvs.eq.1)read(1,501)matname(i),matdef(i)
  500 format(2x,a8,a35)
  501 format(2x,a8,35x,a35)
    4 continue
      print card
      if(nlist.eq.1)stop
c modtype est le mom du modèle (8 caractères) cadré à droite
      do 1005 i=1,neq
 1005 modtype(i)='   L-2.1'
c Initialiser les 9 titres des statistiques pour Beta ou Delta.
      if(nvs.eq.0)then
         stitle(1)='Coefficient BETA'
         stitle(2)='Dérivée de Y (valeur calculée)'
         stitle(3)='Dérivée de E(Y)'
         stitle(4)='Élasticité de Y (valeur calculée)'
         stitle(5)='Élasticité de E(Y)'
         stitle(6)='Corrélation (Y,X_k) unités origin.'
         stitle(7)='Corrélation (Y,X_k) transf. Box-Cox'
         stitle(8)='Taux Marg. de Subst.(d X_ref/d X_k)'
         stitle(9)='Stat. t conditionnelle pour BETA'
      else
         stitle(1)='BETA coefficient'
         stitle(2)='Derivative of Y (sample value)'
         stitle(3)='Derivative of E(Y)'
         stitle(4)='Elasticity of Y (sample value)'
         stitle(5)='Elasticity of E(Y)'
         stitle(6)='Correlation (Y,X_k) original units'
         stitle(7)='Correlation (Y,X_k) Box-Cox transf.'
```

```
         stitle(8)='Marg. Rate of Subst.(d X_ref/d X_k)'
         stitle(9)='Conditional t-statistic for BETA'
      endif
c*******************************************************
c BEGINNING OF LOOP 30 (i=1,neq) for the 'neq' variants.
c Read the results of the variants on UNIT=2.
c*******************************************************
      do 30 i=1,neq
      npar(i)=0
      read 29,fname
      open(unit=2,file=fname)
      read(2,1020)namy(i),varname,ivernum(i)
 1020 format(a8,2x,a9,i4)
      nvtname(i)=adjustr(varname)

      call tasser(namy(i),8)
      read(2,1021)nobs,nvaxz,nvax(i),norder(i),nvaz(i),nlamz(i),
     +nlamy(i),nlamx(i)
 1021 format(8i5)
      nvi(i)=nvax(i)+1
      nb1=nvax(i)+1
c   Lire les noms des variables independantes X, les groupes de Lambda(x),
c   les coefficients Betas, les types de Lambda(x), les 2 t-students
(inconditionnel
C   et conditionnel), les 2 correlations entre Y et X_k (originaux et
transformes),
c   les 2 types de derivees, et les 2 types d'elasticites pour chaque passe (ou
variante).
      do 40 j=1,nb1
      npar(i)=npar(i)+1
      read(2,1420)name,numlx,b,ntypelx,stu,stuc,(coryx(j1),j1=1,2),
     +(deryx(j1),ela(j1),j1=1,2)
 1420 format(a8,2x,i2,d20.12,a3,4f12.4/4d20.12)
c Store the derivative of Y sample with respect to X_reference, to be used in
c the computation of the MRS of X_k with X_reference, for each variant.
      if(name.eq.xrefname(i))deryxref(i)=deryx(1)
      do 50 k=1,nvar
      if(name.eq.namv(k))go to 41
   50 continue
      print 1031,name,ivernum(i)
 1031 format(///5xa8" : This variable used in Variant no.",i4,
     +"does not appear in the file containing the list of definitions"/
     +"and categories of the independent variables X")
      stop
   41 continue
c Store 9 values which correspond to a Beta-coefficient, 2 derivatives,
c 2 elasticities, 2 correlations between Y and X_k (original and transformed),
c a MRS of X_k with X_reference and a conditional student-t, for each variable
c of an equation in the big array X(k,i,j) where k is the k-th variable,
c i is the i-th equation and j is the j-th value.
      x(k,i,1)=b
      do 42 j1=2,3
   42 x(k,i,j1)=deryx(j1-1)
      do 43 j1=4,5
   43 x(k,i,j1)=ela(j1-3)
      x(k,i,6)=coryx(1)
      x(k,i,7)=coryx(2)
```

```
c Note: At this step, x(k,i,8) contains only the numerator of the MRS,
c       which corresponds to the derivative of Y sample with respect to X_k.
      x(k,i,8)=deryx(1)
      x(k,i,9)=stuc
      nbeta(k,i)=1
      nty(k,i)=ntypelx
      ngr(k,i)=numlx
   40 continue

      nro=norder(i)
      if(nro.gt.0)then
         do 140 j=1,nro
  140    read(2,1291)imatname(j)
 1291    format(a8)
         do 61 j=1,nro
         do 44 k=1,nmatr
         if(imatname(j).eq.matname(k))go to 45
   44    continue
         print 1033,name,ivernum(i)
 1033    format(///5xa8" : This matrix name used in Variant no.",i4,
     +"does not appear in the file containing the list of definitions"/
     +"of the matrices R")
         stop
   45    continue

c Read the estimated Rho's (The Rho's are always estimated, not fixed).
c Use only "stu" which is the unconditional t-stdent.
         ntrho(k,i)=1
         read(2,1292)l,rho(k,i,1),stu,stucond
 1292    format(3x,i1,d20.12,3x,2f12.4)
         nroest(i)=nroest(i)+1
         npar(i)=npar(i)+1
         rho(k,i,2)=stucond
61    continue
c Read the fixed and estimated Pi's.
c Use only "student" and "stud1" which are the unconditional t-student's.
c (ntpi=0 if Pi is fixed, ntpi=1 if Pi is estimated)
            do 73 j=1,nro
         do 161 k=1,nmatr
      if(imatname(j).eq.matname(k))then
         read(2,1293)l,pi(k,i,1),ntpi(k,i),student,stud1,stucond,
     +   stu1cond
 1293    format(3x,i1,d20.12,i3,4f12.4)
         if(ntpi(k,i).eq.1)then
            npiest(i)=npiest(i)+1
            npar(i)=npar(i)+1
            pi(k,i,2)=stucond
            pi(k,i,3)=stu1cond
         else
            npifix(i)=npifix(i)+1
         endif
      endif
  161    continue
  73        continue
      endif

c   Read the fixed and estimated Delta(z)'s
```

```
      nd=nvaz(i)
      if(nd.gt.0)then
c name  = name of the heteroskedasticity variable associated with a Delta.
c numlz = index of the Lambda(z) associated with the heteroskedasticity
c         variable.
c delta = value of Delta.
c ntype = 0 if Delta is fixed.
c       = 1 if Delta is estimated.
         do 70 j=1,nd
         read(2,1430)name,numlz(j),delta,ntype,student,stucond,
     +   (coryz(j1),j1=1,2),smpderz(j),smpelaz(j),expderz(j),expelaz(j)
 1430    format(a8,2x,i2,d20.12,i3,4f12.4/4d20.12)
         do 71 k=1,ndt
         if(name.eq.namz(k))go to 72
   71    continue
         print 1032,name,ivernum(i)
 1032    format(///5xa8" : This variable used in Variant no.",i4,/
     +   "in the heteroskedasticity function does not appear in"/
     +   "the second input card")
         stop
c Store the coefficient Delta, the 2 derivatives of Y with respect to Z_m,
c the 2 elasticities, the 2 correlations between Y and Z_m (original and
transformed)
c the derivative of Y with respect to Z_m and the conditional t-statistic.
   72    vd(k,i,1)=delta
         vd(k,i,2)=smpderz(j)
         vd(k,i,3)=expderz(j)
         vd(k,i,4)=smpelaz(j)
         vd(k,i,5)=expelaz(j)
         vd(k,i,6)=coryz(1)
         vd(k,i,7)=coryz(2)
         vd(k,i,8)=smpderz(j)
         vd(k,i,9)=stucond
         if(ntype.eq.1)then
            ndelest(i)=ndelest(i)+1
            npar(i)=npar(i)+1
            ntdel(k,i)=1
         else
            ndelfix(i)=ndelfix(i)+1
            ntdel(k,i)=0
         endif
c   Read the fixed and estimated Lambda(z)'s
c l      = index of Lambda(z).
c vlamz  = value of Lambda(z).
c ntlz   = 0 if Lambda(z) is fixed.
c        = 1 if Lambda(z) is estimated.
         read(2,1440)l,vlamz,ntlz(k,i),student,stud1
 1440    format(10x,i2,d20.12,i3,2f12.4)
         vlz(k,i,1)=vlamz
         vlz(k,i,2)=student
         vlz(k,i,3)=stud1
         if(ntlz(k,i).eq.1)then
            nlzest(i)=nlzest(i)+1
            npar(i)=npar(i)+1
         else
            nlzfix(i)=nlzfix(i)+1
         endif
```

```
   70    continue
      endif
c Read the fixed and estimated Lambda(y) and Lambda(x)'s.
      if(nlamy(i).eq.1)then
c numlxy = 0 if Lambda(y) is not set equal to a Lambda(x).
c        = a positive index of the Lambda(x) set equal to Lambda(y).
c vlamy  = value of Lambda(y).
c ntype  = 0 if Lambda(y) is fixed.
c        = 1 if Lambda(y) is estimated.
         read(2,1440)numlxy,vlamy,ntype,student,stud1
         vly(i,1)=vlamy
         vly(i,2)=student
         vly(i,3)=stud1
         if(ntype.eq.1)then
            ntly(i)=1
            if(numlxy.eq.0)then
                   nlyestdis(i)=1
                 npar(i)=npar(i)+1
              else
                   nlyestcon(i)=1
            endif
         else
              nlyfix(i)=1
            ntly(i)=0
         endif
      endif

      if(nlamx(i).gt.0)then
c k      = index of Lambda(x).
c vlamx  = value of Lambda(x).
c ntype  = 0 if Lambda(x) is fixed.
c        = 1 if Lambda(x) is estimated.
         do 100 j=1,nlamx(i)
         read(2,1440)k,vlamx,ntype,student,stud1
         vlx(j,i,1)=vlamx
         vlx(j,i,2)=student
         vlx(j,i,3)=stud1
         if(ntype.eq.1)then
            nlxest(i)=nlxest(i)+1
            ntlx(j,i)=1
            npar(i)=npar(i)+1
         else
            nlxfix(i)=nlxfix(i)+1
            ntlx(j,i)=0
         endif
  100    continue
      endif

c Read the values: log-likelihood function, 2 Pearson R2's, 2 Pseudo-L-R2's,
c                  probability of Y to be at the (lower or upper) limit,
c                  mean of Y observed (ymean),
c                  and mean of E(Y) estimated (eymean),
      read(2,1480)vlik(i),r2p(i),r2pa(i),r2(i),r2bar(i),plim(i)
 1480 format(6f12.3)
      read(2,1481)ymean(i),eymean(i)
 1481 format(2d20.12)
      close(2)
```

```
   30 continue
c*****************
c   END OF LOOP 30
c*****************
      print 1901
c******************************************************************************
c PART I of TABLEX: Regression Coefficients, Partial Derivatives and
c                   Elasticities of Y and E(y), with respect to X_k,
c                   Correlations between Y and X_k (original and transformed)
c                   MRS (Marginal Rate of Substitution) of X_k with X_reference
c                   and conditional t-statistic.
c******************************************************************************
      ns=47+neq*10
      encode(20,1910,ifmt)ns
 1910 format('(/1x,',i3,"('='))")
      print ifmt

      if(nvs.eq.0)then
            print 1911,(modtype(i),i=1,neq)
 1911       format(1x,'PARTIE I. Statistiques présentées'/
     +      3x,'Type de modèle',t49,10(2x,a8))
            print 1912,(nvtname(i),i=1,neq)
 1912       format(3x,'Nom de la variante',t49,10(1x,a9))
            print 1913,(ivernum(i),i=1,neq)
 1913       format(3x,'Numéro de version de la variante',t49,10(6x,i4))
            print 1914,(namy(i),i=1,neq)
 1914       format(3x,'Variable dépendante dans la variante',t49,10(2x,a8))
            if(nstat(8).ne.0)print 1915,(namxref(i),i=1,neq)
 1915       format(3x,'Variable de référence X_ref pour T.M.S.',t49,
     +      10(2x,a8))
      else
            print 1921,(modtype(i),i=1,neq)
 1921       format(1x,'PART I. Presented statistics'/
     +      3x,'Model type',t49,10(2x,a8))
            print 1922,(nvtname(i),i=1,neq)
 1922       format(3x,'Name of the variant',t49,10(1x,a9))
            print 1923,(ivernum(i),i=1,neq)
 1923       format(3x,'Version number of the variant',t49,10(6x,i4))
            print 1924,(namy(i),i=1,neq)
 1924       format(3x,'Dependent variable in the variant',t49,10(2x,a8))
            if(nstat(8).ne.0)print 1925,(namxref(i),i=1,neq)
 1925       format(3x,'Reference variable X_ref for M.R.S.',t49,10(2x,a8))

      end if
      print ifmt

c***********************************************************
c DEBUT DE LA BOUCLE 201 pour les Variables independantes X
c***********************************************************
      jj=0
      rewind 1
      do 201 i=1,nbloc
      if(nvs.eq.0)read(1,1001)num,nambl,idef
      if(nvs.eq.1)read(1,1002)num,nambl,idef
      decode(35,202,idef)icar
  202 format(35a1)
      do 203 j=1,35
```

```
      nk=35-j+1
      if(icar(nk).ne.' ')go to 204
  203 continue
  204 continue
      j1=0
  200 continue
      if(nvs.eq.0)read(1,1001)num,namx,idef
      if(nvs.eq.1)read(1,1002)num,namx,idef
      if(namx.eq.' ')go to 201
      jj=jj+1
      ns=0
      do 210 k=1,neq
  210 ns=ns+nbeta(jj,k)
      if(ns.eq.0)go to 200
      j1=j1+1
      if(j1.ne.1)go to 205
      nk8=nk+8
      print 111
  111 format(' ')
      encode(40,207,fmt)nk8
  207 format('(1x,',i2,'("-"))')
      print fmt
      encode(12,2021,alpha)nk
 2021 format('(1x,a8,',i2,'a1)')
      print alpha,nambl,(icar(kk),kk=1,nk)
      print fmt
  205 continue
c Convertir les valeurs numeriques des coefficients Betas, des derivees,
c des elasticites, des correlations et le taux marginal de substitution
c en caracteres.
      do 220 k=1,neq
      do 214 j=1,9
  214 alphax(k,j)=' '
      if(nbeta(jj,k).eq.0)go to 220
c Coefficient Beta et 2 Derivees
      do 211 j=1,3
  211 encode(10,219,alphax(k,j))x(jj,k,j)
  219 format(d10.2)
c 2 Elasticites et 2 correlations
      do 212 j=4,7
  212 encode(10,221,alphax(k,j))x(jj,k,j)
  221 format(f10.3)
c Taux marginal de substitution
      encode(10,219,alphax(k,j))x(jj,k,8)/deryxref(k)
  220 continue

c Convertir la valeur numerique de la statistique t en caracteres.
      do 230 k=1,neq
      if(nbeta(jj,k).ne.0)encode(10,231,alphax(k,9))x(jj,k,9)
  231 format(' ('f7.2')')
  230 continue

c Imprimer la définition et le nom de la variable explicative.
      print 2020,idef,namv(jj)
 2020 format(/3x,a35,2x,a8,10a10)
c Souligner les Quasi-dummies (numv=2) et les dummies (numv=3)
      do 410 k=1,8
```

```
  410 under(k)=' '
      if(numv(jj).gt.1)call uline(numv(jj),namv(jj),under)
      if(numv(jj).eq.2)print 226,under
      if(numv(jj).eq.3)print 227,under
  226 format(3x,'(Quasi-dummy)',t41,8a1)
  227 format(3x,'(Dummy)',t41,8a1)

c Imprimer le coefficient Beta, les 2 derivees, les 2 elasticites,
c les 2 correlations,le taux marginal de substitution et
c la statistique t conditionnelle selon les options choisies
c dans nstat(i), i=1,9.
      do 232 j=1,9
      if(nstat(j).gt.0)print 222,stitle(j),(alphax(k,j),k=1,neq)
  222 format(5x,a35,8x,10a10)
  232 continue

c Imprimer le type et le groupe de Lambda(x) associes a chaque variable.
      do 240 k=1,neq
      alpha(k)=' '
      if(nty(jj,k).eq.' nl')go to 240
      if(nbeta(jj,k).ne.0)then
         if(ngr(jj,k).lt.10)then
            encode(10,241,alpha(k))ngr(jj,k)
  241       format('    LAM ',i1)
         else
            encode(10,242,alpha(k))ngr(jj,k)
  242       format('   LAM ',i2)
         endif
      endif
  240 continue
      print 1222,(alpha(k),k=1,neq)
 1222 format(48x,10a10)
      go to 200
  201 continue
      close(1)
c*******************************************************
c FIN DE LA BOUCLE 201 pour les Variables independantes X
c*******************************************************
c*********************************************************************
c PART I : HETEROSCEDASTICITE
c          Imprimer le coefficient Delta, les 2 derivees, les 2 elasticites,
c          les 2 correlations, le Taux Marginal de Substitution de Z_m avec
c          X_reference et le t conditionnel associes aux Variables Z.
c*********************************************************************
      if(ndt.gt.0)then
         if(nvs.eq.0)then
              print 260
  260       format(/1x'Hétéroscédasticité'/1x,18(1h-))
            stitle(1)='Coefficient DELTA'
            stitle(6)='Corrélation (Y,Z_m) unités origin.'
            stitle(7)='Corrélation (Y,Z_m) transf. Box-Cox'
            stitle(8)='Taux Marg. de Subst.(d X_ref/d Z_m)'
            stitle(9)='Stat. t conditionnelle pour DELTA'
         else
              print 261
  261       format(/1x'Heteroskedasticity'/1x,18(1h-))
            stitle(1)='DELTA coefficient'
```

```
            stitle(6)='Correlation (Y,Z_m) original units'
            stitle(7)='Correlation (Y,Z_m) Box-Cox transf.'
            stitle(8)='Marg. Rate of Subst.(d X_ref/d Z_m)'
            stitle(9)='Conditional t-statistic for DELTA'
         endif
         do 250 i=1,ndt
         do 251 j=1,neq
         do 252 k=1,9
  252    alphax(j,k)=' '
         if(ntdel(i,j).ge.0)then
            do 253 k=1,3
  253       encode(10,219,alphax(j,k))vd(i,j,k)
            do 254 k=4,7
  254       encode(10,221,alphax(j,k))vd(i,j,k)
            encode(10,219,alphax(j,8))vd(i,j,8)/deryxref(j)
            if(ntdel(i,j).eq.1)encode(10,231,alphax(j,9))vd(i,j,9)
         endif
  251    continue
         print 2020,idefz(i),namz(i)
         do 255 k=1,9
  255    if(nstat(k).gt.0)print 222,stitle(k),(alphax(j,k),j=1,neq)
  250    continue
      endif
c
c************************************************************
c PART II of TABLEX: Parameters (Box-Cox, Heteroskedasticity,
c                    Directed Autocorrelation)
c************************************************************
      print ifmt
      if(nvs.eq.0)then
            print 1926,(modtype(i),i=1,neq)
 1926       format(1x,'PARTIE II. Paramètres'/12x,
     +    'Stat. t non conditionnelle (=0) [=1]'/
     +      3x,'Type de modèle',t49,10(2x,a8))
            print 1912,(nvtname(i),i=1,neq)
            print 1913,(ivernum(i),i=1,neq)
            print 1914,(namy(i),i=1,neq)
      else
            print 1927,(modtype(i),i=1,neq)
 1927       format(1x,'PART II. Parameters'/10x,
     +    't-statistic unconditional (=0) [=1]'/
     +      3x,'Model type',t49,10(2x,a8))
            print 1922,(nvtname(i),i=1,neq)
            print 1923,(ivernum(i),i=1,neq)
            print 1924,(namy(i),i=1,neq)
      end if
      print ifmt
c*********************************
c   Box-Cox (Lambda(y) et Lambda(x)
c*********************************
c   nsy    = Nombre total de Lambda(y) fixes et estimes dans les NEQ equations.
c   nsx    = Nombre total de Lambda(x) fixes et estimes dans les NEQ equations.
c   nleq(i) = Nombre total de Lambda(x) fixes et estimes dans les NEQ equations
c             pour le i-eme Lambda(x)
      nsy=0
      do 350 j=1,neq
  350 if(ntly(j).ge.0)nsy=nsy+1
```

```
      do 340 i=1,10
      nleq(i)=0
      do 340 j=1,neq
      if(ntlx(i,j).ge.0)then
         nsx=nsx+1
         nleq(i)=nleq(i)+1
      endif
  340 continue

c Print the fixed and estimated Lambda(y) and Lambda(x)'s.
c   ntot = Nombre total de Lambda(y) et Lambda(x) fixes et estimes
c          dans les NEQ equations.

      ntot=nsy+nsx
      if(ntot.gt.0)then
         if(nvs.eq.0)print 3008
         if(nvs.eq.1)print 3001
 3008    format(/1x'Transformations de Box-Cox'/1x26('-'))
 3001    format(/1x'Box-Cox Transformations'/1x23('-'))

         if(nsy.gt.0)then
 3003       format(/3x'LAMBDA(Y)',t49,10a10)
            do 353 k=1,neq
            alpha(k)=' '
            if(ntly(k).ge.0)encode(10,221,alpha(k))vly(k,1)
  353       continue
            print 3003,(alpha(k),k=1,neq)
            do 355 j=2,3
            do 354 k=1,neq
            alpha(k)=' '
            if(j.eq.2)then
               if(ntly(k).eq.1)then
                  encode(10,231,alpha(k))vly(k,2)
               else if (ntly(k).eq.0) then
                  alpha(k)='      Fixe'
                  if(nvs.eq.1)alpha(k)='     Fixed'
               endif
            else
               if(ntly(k).eq.1)then
                        if(dabs(vly(k,3)).lt.999)then
                              encode(10,233,alpha(k))vly(k,3)
  233                         format(' ['f7.2']')
                        else
                              encode(10,1233,alpha(k))vly(k,3)
 1233                         format(' ['f7.0']')
                        endif
                   endif
            endif
  354       continue
            print 1222,(alpha(k),k=1,neq)
  355       continue
         endif

         if(nsx.gt.0)then
            do 342 i=1,10
            if(nleq(i).gt.0)then
               do 343 k=1,neq
```

```
               alpha(k)=' '
               if(ntlx(i,k).ge.0)encode(10,221,alpha(k))vlx(i,k,1)
  343          continue
               if(nvs.eq.0)print 3011,i,(alpha(k),k=1,neq)
               if(nvs.eq.1)print 3002,i,(alpha(k),k=1,neq)
 3011          format(/3x'LAMBDA(X) - Groupe 'i1,t49,10a10)
 3002          format(/3x'LAMBDA(X) - Group 'i1,t49,10a10)

               do 345 j=2,3
               do 344 k=1,neq
               alpha(k)=' '
               if(j.eq.2)then
                  if(ntlx(i,k).eq.1)then
                       encode(10,231,alpha(k))vlx(i,k,2)
                  else if (ntlx(i,k).eq.0)then
                     alpha(k)='      Fixe'
                     if(nvs.eq.1) alpha(k)='     Fixed'
                  endif
               else
                    if(ntlx(i,k).eq.1)then
                              if(dabs(vlx(i,k,3)).lt.999)then
                                    encode(10,233,alpha(k))vlx(i,k,3)
                              else
                                    encode(10,1233,alpha(k))vlx(i,k,3)
                              endif
                        endif
               endif
  344          continue
  345          print 1222,(alpha(k),k=1,neq)
            endif
  342       continue
         endif
      endif
c*********************
c   Heteroskedasticity
c*********************
      if(ndt.gt.0)then
c Print the fixed and estimated Lambda(z)'s.
         if(nvs.eq.0)print 3007
         if(nvs.eq.1)print 3004
 3007    format(/1x'Hétéroscédasticité'/1x18('-'))
 3004    format(/1x'Heteroskedasticity'/1x18('-'))
         do 362 i=1,ndt
         print 361,idefz(i),namz(i)
  361    format(/3x,a35,2x,a8)

         do 373 k=1,neq
         alpha(k)=' '
         if(ntlz(i,k).ge.0)encode(10,221,alpha(k))vlz(i,k,1)
  373    continue
         print 3006,(alpha(k),k=1,neq)
 3006    format(/6x'LAMBDA(Z)',t49,10a10)
         do 375 j=2,3
         do 374 k=1,neq
         alpha(k)=' '
         if(j.eq.2)then
            if(ntlz(i,k).eq.1)then
```

```
               encode(10,231,alpha(k))vlz(i,k,2)
            else if (ntlz(i,k).eq.0)then
               alpha(k)='      Fixe'
               if(nvs.eq.1) alpha(k)='     Fixed'
            endif
         else
            if(ntlz(i,k).eq.1)then
                        if(dabs(vlz(i,k,3)).lt.999)then
                              encode(10,233,alpha(k))vlz(i,k,3)
                        else
                              encode(10,1233,alpha(k))vlz(i,k,3)
                        endif
              endif
         endif
  374    continue
  375    print 1222,(alpha(k),k=1,neq)
  362    continue
      endif
c*************************
c  Directed Autocorrelation
c*************************
c nr(i) = Nombre de Rho estimes dans les NEQ equations pour la ieme matrice R.
      ns=0
      do 300 i=1,nmatr
      nr(i)=0
      do 300 j=1,neq
      if(ntrho(i,j).eq.1)then
         nr(i)=nr(i)+1
         ns=ns+1
      endif
  300 continue

      if(ns.ne.0)then
         if(nvs.eq.0)then
            print 2999
 2999       format(/1x'Autocorrélation Dirigée'/1x24('-'))
         else
            print 3000
 3000       format(/1x'Directed Autocorrelation'/1x24('-'))
         endif
         do 310 i=1,nmatr
         if(nr(i).gt.0)then
            print 2020,matdef(i)
c Print RHO and t-statistic. RHO is always estimated.
            do 320 k=1,neq
            alpha(k)=' '
            if(ntrho(i,k).eq.1)encode(10,221,alpha(k))rho(i,k,1)
  320       continue
            print 3009,(alpha(k),k=1,neq)
 3009       format(/6x'RHO',t49,10a10)
            do 330 k=1,neq
            alpha(k)=' '
            if(ntrho(i,k).eq.1)encode(10,231,alpha(k))rho(i,k,2)
  330       continue
            print 1222,(alpha(k),k=1,neq)
c Print PI and two t-statistics. Pi can be fixed or estimated.
            do 331 k=1,neq
```

```
          alpha(k)=' '
          if(ntpi(i,k).ne.-1)encode(10,221,alpha(k))pi(i,k,1)
 331      continue
          print 3010,(alpha(k),k=1,neq)
3010      format(/6x'PI',t49,10a10)

          do 332 j=2,3
          do 333 k=1,neq
          alpha(k)=' '
          if(j.eq.2)then
             if(ntpi(i,k).eq.1)then
                  encode(10,231,alpha(k))pi(i,k,2)
             elseif(ntpi(i,k).eq.0)then
                alpha(k)='      Fixe'
                if(nvs.eq.1) alpha(k)='     Fixed'
             endif
          else
               if(ntpi(i,k).eq.1)then
                     if(dabs(pi(i,k,3)).lt.999)then
                           encode(10,233,alpha(k))pi(i,k,3)
                     else
                           encode(10,1233,alpha(k))pi(i,k,3)
                     endif
                 endif
          endif
 333      continue
 332      print 1222,(alpha(k),k=1,neq)
       endif
 310   continue
    endif
c***************************************
c Part III of TABLEX: General Statistics
c***************************************
    print ifmt
    if(nvs.eq.0)then
          print 1928,(modtype(i),i=1,neq)
1928      format(1x,'PARTIE III. Statistiques générales'/
    +     3x,'Type de modèle',t49,10(2x,a8))
          print 1912,(nvtname(i),i=1,neq)
          print 1913,(ivernum(i),i=1,neq)
          print 1914,(namy(i),i=1,neq)
          print ifmt
          print 6000,(vlik(i),i=1,neq)
6000      format(/1x'Log de vraisemblance't49,10f10.2)
          print 6100,(npar(i)-1,i=1,neq)
6100      format(/1x'Degrés de liberté't49,10i10)
          print 6009,(r2p(i),i=1,neq)
6009      format(/1x'Pearson R2',t49,10f10.3)
          print 6018,(r2pa(i),i=1,neq)
6018      format(1x'Pearson R2 ajusté pour Degrés de liberté',t49,
    +10f10.3)
          print 6010,(r2(i),i=1,neq)
6010      format(/1x'Pseudo-(L)-R2't49,10f10.3)
          print 6020,(r2bar(i),i=1,neq)
6020      format(1x'Pseudo-(L)-R2 ajusté pour Degrés de liberté't49,
    +10f10.3)
          print 6050,(plim(i),i=1,neq)
```

```
 6050       format(/1x'Probabilité moyenne (Y=observation limite)'t49,
     +      10f10.3)
            print 6044,(nobs,i=1,neq)
 6044       format(/1x"Échantillon - Nombre d'observations"t49,10i10)

            print 6040
 6040       format(/1x'Nombre total de paramètres fixés ou estimés:')
            print 6141,(nvi(i)-1,i=1,neq)
 6141       format(13x,'- Beta .Estimé',t49,10i10)
            print 61411,(ncst(i),i=1,neq)
61411       format(20x,'.Constante',t49,10i10)

            print 6142,(nlyfix(i),i=1,neq)
 6142       format(13x,'- Lambda(y)'/20x,'.Fixé',t49,10i10)
            print 61421,(nlyestcon(i),i=1,neq)
61421       format(20x,'.Estimé .Contraint',t49,10i10)
            print 61422,(nlyestdis(i),i=1,neq)
61422       format(28x,'.Distinct',t49,10i10)

            print 61430,(nlxfix(i),i=1,neq)
61430       format(13x'- Lambda(x)'/20x,'.Fixé',t49,10i10)
            print 61431,(nlxest(i),i=1,neq)
61431       format(20x,'.Estimé',t49,10i10)

            print 6143,(ndelfix(i),i=1,neq)
 6143       format(13x'- Hétéroscédasticité'/16x,'Delta'/20x,'.Fixé',t49,
     +10i10)
            print 61432,(ndelest(i),i=1,neq)
61432       format(20x,'.Estimé',t49,10i10)

            print 6144,(nlzfix(i),i=1,neq)
 6144       format(16x'Lambda(z)'/20x,'.Fixé',t49,10i10)
            print 61441,(nlzest(i),i=1,neq)
61441       format(20x,'.Estimé',t49,10i10)

            print 6145,(nroest(i),i=1,neq)
 6145       format(13x'- Autocorrélation'/16x,'Rho .Estimé',t49,10i10)

            print 6146,(npifix(i),i=1,neq)
 6146       format(16x,'Pi  .Fixé',t49,10i10)
            print 61460,(npiest(i),i=1,neq)
61460       format(20x,'.Estimé',t49,10i10)

            print 6147,(nlyfix(i)+nlxfix(i)+ndelfix(i)+nlzfix(i)+npifix(i)
     +,i=1,neq)
 6147       format(13x'- Total'/20x,'.Fixé',t49,10i10)
            print 6148,(npar(i),i=1,neq)
 6148       format(20x,'.Estimé',t49,10i10)

            print 6051,(ymean(i),i=1,neq)
 6051       format(/1x'Moyenne de Y observé't49,10d10.2)
            print 6057,(eymean(i),i=1,neq)
 6057       format(/1x'Moyenne de E(Y) estimé't49,10d10.2)
      else
            print 1929,(modtype(i),i=1,neq)
 1929       format(1x,'PART III. General statistics'/
     +      3x,'Model type',t49,10(2x,a8))
```

```
          print 1922,(nvtname(i),i=1,neq)
          print 1923,(ivernum(i),i=1,neq)
          print 1924,(namy(i),i=1,neq)
          print ifmt
          print 6001,(vlik(i),i=1,neq)
 6001     format(/1x'Log-likelihood't49,10f10.2)
          print 6101,(npar(i)-1,i=1,neq)
 6101     format(/1x'Degrees of freedom't49,10i10)
          print 6009,(r2p(i),i=1,neq)
          print 6019,(r2pa(i),i=1,neq)
 6019     format(1x'Pearson R2 adjusted for Degrees of freedom',t49,
     +10f10.3)
          print 6010,(r2(i),i=1,neq)
          print 6021,(r2bar(i),i=1,neq)
 6021     format(1x'Pseudo-(L)-R2 adjusted for Degrees of freedom't49,
     +10f10.3)
          print 6121,(plim(i),i=1,neq)
 6121     format(/1x'Average probability (Y=limit observation)'t49,
     +    10f10.3)
          print 6045,(nobs,i=1,neq)
 6045     format(/1x'Sample     - Number of observations't49,10i10)

          print 6340
 6340     format(/1x'Total number of fixed or estimated parameters:')
          print 6341,(nvi(i)-1,i=1,neq)
 6341     format(12x'- Beta .Estimated't49,10i10)
          print 63411,(ncst(i),i=1,neq)
63411     format(19x,'.Constant',t49,10i10)

          print 6342,(nlyfix(i),i=1,neq)
 6342     format(12x,'- Lambda(y)'/19x,'.Fixed',t49,10i10)
          print 63421,(nlyestcon(i),i=1,neq)
63421     format(19x,'.Estimated .Constrained',t49,10i10)
          print 63422,(nlyestdis(i),i=1,neq)
63422     format(30x,'.Distinct',t49,10i10)

          print 63430,(nlxfix(i),i=1,neq)
63430     format(12x'- Lambda(x)'/19x,'.Fixed',t49,10i10)
          print 63431,(nlxest(i),i=1,neq)
63431     format(19x,'.Estimated',t49,10i10)

          print 6343,(ndelfix(i),i=1,neq)
 6343     format(12x'- Heteroskedasticity'/15x,'Delta'/19x,'.Fixed',t49,
     +10i10)
          print 63432,(ndelest(i),i=1,neq)
63432     format(19x,'.Estimated',t49,10i10)

          print 6344,(nlzfix(i),i=1,neq)
 6344     format(15x'Lambda(z)'/19x,'.Fixed',t49,10i10)
          print 63441,(nlzest(i),i=1,neq)
63441     format(19x,'.Estimated',t49,10i10)

          print 6345,(nroest(i),i=1,neq)
 6345     format(12x'- Autocorrelation'/15x,'Rho .Estimated't49,10i10)

          print 6346,(npifix(i),i=1,neq)
 6346     format(15x,'Pi  .Fixed't49,10i10)
```

```
            print 6347,(npiest(i),i=1,neq)
6347        format(19x,'.Estimated't49,10i10)

            print 6348,(nlyfix(i)+nlxfix(i)+ndelfix(i)+nlzfix(i)+npifix(i)
    +,i=1,neq)
6348        format(12x'- Total'/19x,'.Fixed',t49,10i10)
            print 6349,(npar(i),i=1,neq)
6349        format(19x,'.Estimated',t49,10i10)

            print 6081,(ymean(i),i=1,neq)
6081        format(/1x'Mean of observed Y't49,10d10.2)
            print 6087,(eymean(i),i=1,neq)
6087        format(/1x'Mean of estimated E(Y)'t49,10d10.2)
      end if
      print ifmt

      stop
      end

      subroutine tasser(namy,n)
      character*8 namy
      character*1 icar(8)
      decode(n,1,namy)(icar(k),k=1,n)
    1 format(8a1)
      do 2 k=1,n
      nk=n-k+1
      if(icar(nk).ne.' ')go to 3
    2 continue
    3 if(k.eq.n)go to 6
      do 4 k=1,nk
      k2=n-k+1
      k1=nk-k+1
    4 icar(k2)=icar(k1)
      k2=n-nk
      do 5 k=1,k2
    5 icar(k)=' '
      encode(n,1,namy)(icar(k),k=1,n)
    6 continue
      return
      end
      subroutine uline(num,nam,under)
      character*8 nam
      character*1 icar(8),under(8)
      decode(8,1,nam)icar
    1 format(8a1)
      do 2 j=1,8
      nk=8-j+1
      if(icar(nk).ne.' ')go to 3
    2 continue
    3 continue
      if(num.eq.3)go to 4
      do 5 j=1,nk
    5 under(j)='-'
      return
    4 do 6 j=1,nk
    6 under(j)='='
      return
```

```
      end

      subroutine tasser(namy,n)
      character*8 namy
      character*1 icar(8)
      decode(n,1,namy)(icar(k),k=1,n)
    1 format(8a1)
      do 2 k=1,n
      nk=n-k+1
      if(icar(nk).ne.' ')go to 3
    2 continue
    3 if(k.eq.n)go to 6
      do 4 k=1,nk
      k2=n-k+1
      k1=nk-k+1
    4 icar(k2)=icar(k1)
      k2=n-nk
      do 5 k=1,k2
    5 icar(k)=' '
      encode(n,1,namy)(icar(k),k=1,n)
    6 continue
      return
      end
      subroutine uline(num,nam,under)
      character*8 nam
      character*1 icar(8),under(8)
      decode(8,1,nam)icar
    1 format(8a1)
      do 2 j=1,8
      nk=8-j+1
      if(icar(nk).ne.' ')go to 3
    2 continue
    3 continue
      if(num.eq.3)go to 4
      do 5 j=1,nk
    5 under(j)='-'
      return
    4 do 6 j=1,nk
    6 under(j)='='
      return
      end
```

4. REFERENCES

All the publications listed below, except for Gaudry (1990), Gaudry *et al.* (1996), Helliwell (1998) and McCallum (1994), are downloadable from the Agora Jules Dupuit website currently at www.e-ajd.net.

Papers using McCallum's 1988 North American trade flow database

Gaudry, M, (2004). The Robust Gravity Form in Transport and Trade Models. Zeitschrift fűr Kanada-Studien 24, 1, Band 44, 144-161.

Gaudry, M., Blum, U. and J. McCallum (1996). A Fist Gross Measure of Unexploited Single Market Integration. In S. Urban, ed., Europe's Challenges , Gabler Verlag, p. 449- 461.

Helliwell, J. F. (1998). How Much Do National Borders Matter? 155 p., Brookings Institution, Washington DC.

McCallum, J. (1995). National Borders Matter: Canada-U.S. Regional Trade Patterns. *American Economic Review* 85, 3, p. 615-623.

Papers using Transport Canada's 1976 Canadian intercity passenger flow database

Gaudry, M., (1990). Three families of mode choice models applied to intercity travel demand with aggregate Canadian data. Discussion paper no 13, John Deutsch Institute for the Study of Economic Policy, Department of Economics, Queen's University, Kingston, 26 p.

Gaudry, M. (1993). Asymmetric Shape and Variable Tail Thickness in Multinomial Probabilistic Responses to Significant Transport Service Level Changes. Publication CRT-893 and Comtois, C. and F. Bingquan, eds. Proceedings of the '92 International Conference on Urban Land Use and Transport System ICULTS '92, Publication CRT-910, Centre de recherche sur les transports, Université de Montréal, April.

Gaudry, M., Mandel, B. and W. Rothengatter (1994, 2008). Introducing Spatial Competition through an Autoregressive Contiguous Distributed (AR-C-D) Process in Intercity Generation-Distribution Models within a Quasi-Direct Format (QDF). Publication CRT-971, Centre de recherche sur les transports, 61 pages, 1994. Revised as Publication AJD-14, Agora Jules Dupuit, Université de Montréal, October 2008.

Liem, T. and M. Gaudry (1994, 1998). QDF: a Quasi-Direct Format used to combine total and mode choice results to obtain modal elasticities and diversion rates. Publication CRT-982, Centre de recherche sur les transports (CRT), Université de Montréal, 21 p., June 1994 and 1998. Also Working Paper N° 9812, Bureau d'économie théorique et appliquée (BETA), Université Louis Pasteur, 1998.

Other

Tran, C.L. and M. Gaudry (2008), "LEVEL: The L-2.1 estimation procedures for BC-DAUHESEQ (Box-Cox Directed AUtoregressive HEteroskedastic Single EQuation) regression". Publication AJD-108, Agora Jules Dupuit, Université de Montréal, April 2008.