Maximum-Likelihood Estimation of Stochastic Frontier Production Functions with Time-Varying Technical Efficiency using the Computer Program, FRONTIER Version 2.0

T. J. Coelli

No. 57 - October 1991

ISSN 0157-0188

ISBN 0 85834 975 2

Maximum-Likelihood Estimation of Stochastic Frontier Production Functions with Time-Varying Technical Efficiency using the Computer Program, FRONTIER Version 2.0

by

T. J. Coelli
Department of Econometrics
University of New England
Armidale, NSW, 2351
Australia

ABSTRACT

Maximum-likelihood estimation of the parameters of a stochastic frontier production function can be a time consuming task. This paper describes a computer program which has been written to provide estimates of the parameters of the model set out in Battese and Coelli (1991). This stochastic frontier production function considers unbalanced panel data with firm effects that are distributed truncated normal and are permitted to vary across time. The computer program permits the estimation of many other models which have appeared in the literature through the imposition of simple restrictions. A three-step method is used to calculate parameter estimates. Asymptotic estimates of standard errors are calculated along with individual and mean estimates of technical efficiency. A short example illustrates the use of the computer program.

1. INTRODUCTION

This paper describes the computer program, FRONTIER, which has been written to obtain maximum likelihood estimates for parameters of a wide variety of stochastic frontier production function models. The paper is divided into sections. Section 2 describes the stochastic frontier production function of Battese and Coelli (1991) and notes the many special cases of this formulation which can be estimated. Section 3 describes the FRONTIER program and Section 4 provides a short example. Some final points are made in Section 5. Two appendices are added, the first summarises key points on the use of the program and also provides a brief explanation of the purposes of each subroutine, while the second appendix contains a listing of the Fortran 77 code.

2. MODEL SPECIFICATION

The stochastic frontier production function, independently proposed by Aigner, Lovell and Schmidt (1977) and Meeusen and van den Broeck (1977), has spawned a body of theoretical and applied literature which is expanding exponentially. I do not attempt to review this literature, but simply refer the reader to the many comprehensive reviews which are available, such as Forsund, Lovell and Schmidt (1980), Schmidt (1986), Bauer (1990) and Battese (1991). The extensive bibliographies of frontier modelling and efficiency analysis provided by Ley (1990) and Beck (1991) may also be of interest.

The computer program, FRONTIER, provides maximum-likelihood estimates of the stochastic frontier production function set out in Battese and Coelli (1991).

$$Y_{it} = f(X_{it};\beta).e^{(V_{it}-U_{it})} \qquad ,i=1,\ldots,N,\ t=1,\ldots,T,$$

where Y_{it} is the production of the i-th firm in the t-th time period,

$f(.)$ is a suitable function (for example the Cobb-Douglas),

X_{it} is the k by 1 vector of input quantities of the i-th firm in the t-th time period,

β is a k by 1 vector of unknown parameters,

V_{it} are i.i.d. $N(0,\sigma_V^2)$,

$U_{it} = (U_i.e^{-\eta(t-T)})$, where η is an unknown parameter and U_i are i.i.d. positive truncations of the $N(\mu,\sigma^2)$ distribution,

and the panel of data need not be complete.

We utilise the parameterization of Battese and Corra (1977) who replace σ_V^2 and σ^2 with $\sigma_S^2=\sigma_V^2+\sigma^2$ and $\gamma=\sigma^2/(\sigma_V^2+\sigma^2)$. This is done with the calculation of the maximum-likelihood estimates in mind. The parameter, γ, must lie between 0 and 1 and thus this range can be searched to provide a good starting value for an iterative maximization process.

The imposition of one or more restrictions upon this model formulation can provide a number of the special cases of this particular model which have appeared in the literature. Setting η to be zero provides the time-invariant model set out in Battese, Coelli and Colby (1989). Furthermore, restricting the formulation to a full (balanced) panel of data gives us the model of Battese and Coelli (1988) and Hughes (1988). The additional restriction of μ equal to

zero reduces the model to Model I in Pitt and Lee (1981). One may add a fourth restriction of T=1 to return to the original formulation of Aigner, Lovell and Schmidt (1977). Obviously a large number of permutations exist. For example, if all these restrictions except μ=0 are imposed, then the model suggested by Stevenson (1980) is obtained.

FRONTIER can estimate all of the above-mentioned models. If the user is unsure as to which formulation is most applicable to a particular study, it is recommended that a number of these models be estimated and that a prefered model be selected using likelihood-ratio tests. One can also test whether any form of stochastic frontier production function is required at all by testing the significance of γ. If the null hypothesis, that γ equals zero, is accepted, then the ordinary least-squares model would be prefered over a stochastic frontier production function.

3. THE FRONTIER PROGRAM

Version 2.0 of FRONTIER is substantially different from the first version documented in Coelli (1989). The major differences are as follows.

- Time-varying technical efficiency is considered.
- Interactive use is now permitted as well as batch use.
- The code has been written for Lahey Fortran 77 for use on a 640k IBM compatible PC.
- The code is distributed on a 360k diskette and includes both the source code as well as a compressed executable file.
- A number of key values such as convergence criterion and iteration limits may be altered without recompiling the code. This is achieved by editing a start-up file named FRONT2.000.
- Numerous small improvements to the code have been made, including the correction of a few minor bugs.

The program is designed to estimate a Cobb-Douglas production function in logarithms.

$$\ln Y_{it} = \beta_0 + \beta_1 . \ln X_{1it} + \ldots + \beta_k . \ln X_{kit} + V_{it} - U_{it}$$

$$, i=1,\ldots,N, \quad t=1,\ldots,T,$$

where k refers to the number of inputs and all other notation is as defined previously.

3.1 FILES NEEDED

The execution of FRONTIER Version 2.0 on an IBM PC generally involves five files:

1) The executable file, FRONT2.EXE
2) The start-up file, FRONT2.000
3) A data file (for example, called test.dta)
4) An instruction file (for example, called test.ins)
5) An output file (for example, called test.out).

The start-up file, FRONT2.000, contains values for a number of key variables such as the convergence criterion, printing flags,etc. This text file may be edited if the user wishes to alter any values. This file is discussed further in Appendix A. The data and instruction files must be created prior to execution. The output file is created by FRONTIER during execution. [Note: A model can be estimated without an instruction file if the program is used interactively.] Examples of a data, instruction and output files are listed in Tables 1, 2 and 3 in the following section.

The program requires that the data be held in an ASCII file and is quite particular about the format. The data must be listed by observation. There must be k+3 columns presented in the following order:

1) Firm number (an integer in the range 1 to N)
2) Period number (an integer in the range 1 to T)
3) Y_{it}
4) X_{1it}
:
:
k+3) X_{kit}

Note that the data must be in original units because the program calculates the logarithms of the output and input data for the estimation of a Cobb-Douglas production function in logarithmic form. The inclusion of any zero or negative numbers in the data will cause the program to stop.[1] The observations can be listed in any order but the columns must be in the stated order. There must be at least one observation on each of the N firms and there must be at least one observation in time period 1 and in time period T.

As noted earlier, the program can receive instructions either from a file or from a terminal. After typing "FRONT2" to begin execution the user is asked whether instructions will come from a file or the terminal. The structure of the instruction file is listed in the next section. If the interactive (terminal) option is selected, then questions will be asked in the same order as they appear in the instruction file.

[1]Functional forms other than Cobb-Douglas may be estimated if the user can 'trick' the program into estimation through suitable transformation of the data columns prior to inclusion in the data file.

Because FRONTIER Version 2.0 was written so that it would run on an IBM PC with 640k RAM, the program now has tighter size limits than previously was the case. The limits are:

100 cross-sections,

20 time periods,

16 regresssor variables (including the constant) and hence 20 parameters.

The code has been rewritten so that these values may be easily altered if the computer on which FRONTIER is loaded has larger (or smaller) limits. The line:

"PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)"

now appears in many program units in the code. The 'search and replace' command in your text editor will permit you to alter this line throughout the program in only a few seconds. The code would then need to be recompiled prior to execution.

3.2 THE THREE-STEP ESTIMATION METHOD

The program will follow a three-step procedure in estimating the maximum-likelihood estimates of the parameters of the stochastic frontier production function.[2] The three steps are:

1) Ordinary Least-Squares (OLS) estimates of the production function are obtained. All β estimators with the exception of the intercept are unbiased.

2) A two-phase grid search of the γ, μ and η parameters is conducted with the β-parameters (excepting β_0) set to the OLS values and the β_0 and σ_s^2 parameters suitably adjusted.

3) The values selected in the grid search are used as starting values in an iterative procedure using a Quasi-Newton method which obtains the final (approximate) maximum-likelihood estimates.

[2] If starting values are specified in the instruction file, the program will skip the first two steps of the procedure.

3.2.1 GRID SEARCH

As mentioned earlier a grid search is conducted across the three parameters, γ, μ and η, if no restrictions are imposed upon the model. The values of β_0 and σ_s^2 are adjusted accordingly during this grid search. With a large data set and a slow computer the grid search may take a number of minutes of computing time to complete. The values of the variables, GRIDNO and GWIDTH, which specify the extent of the grid search are listed in the start-up file, FRONT2.000. The grid search is as follows:

γ : from 0.1 to 0.9 in steps of GRIDNO

μ : from $-2.\sigma_{ols}$ to $2.\sigma_{ols}$ in steps of $2.\sigma_{ols}$.GRIDNO

η : from -GWIDTH to GWIDTH in steps of 2.GWIDTH.GRIDNO

where σ_{ols} is the standard error of the OLS regression conducted in step (1) and the values of GRIDNO and GWIDTH are set in the file FRONT2.0C0 to:

GRIDNO = 0.1

GWIDTH = 1.0.

These values may be altered by the user if desired.

With the above values the grid search will involve (9 by 11 by 11) = 1089 function evaluations. If the width of the search on η is altered (by changing GWIDTH) this will not affect the number of function evaluations conducted. If the size of the steps are altered (by changing GRIDNO), then the number of function evaluations will change. For example, reducing GRIDNO from 0.1 to 0.05 will result in (17 by 21 by 21) = 7497 function evaluations. Hence reducing GRIDNO by half has resulted in approximately (2 by 2 by 2) = 8 times the number of function evaluations in the grid search. If μ and/or η are restricted to be zero the number of function evaluations in the grid search will be reduced substantially.

If the variable, IGRID2, in FRONT2.000 is set to 1 (instead of 0) then a second-phase grid search will be conducted around the values obtained in the first phase. The width of this grid search on all three parameters will be GRIDNO/2 either side of the first-phase estimates in steps of GRIDNO/5. Hence, there will be an extra (6 by 6 by 6) = 216 function evaluations. The value of IGRID2 may be set to 0 in FRONT2.000 if the user does not require this extra degree of accuracy on the grid search.

3.2.2 ITERATIVE MAXIMIZATION PROCEDURE

The first-order partial derivatives of the log-likelihood function of the stochastic frontier production function are lengthy expressions. These are presented in the appendix of Battese and Coelli (1991). Many of the gradient methods of finding maximum-likelihood estimates, such as the Newton-Raphson method, require the second partial derivatives to be calculated. Hence we decided to use Quasi-Newton methods which only require the vector of first partial derivatives. The Davidon-Fletcher-Powell method was selected as it appears to have been used successfully in a wide range of econometric applications and was also recommended by Pitt and Lee (1981). For a general discussion of the relative merits of a number of Newton and Quasi-Newton methods, see Himmelblau (1972), which also provides a description of the mechanics (along with Fortran the code) of a number of the more popular methods. The structure of the subroutines, MINI, SEARCH, ETA and CONVRG, used in FRONTIER are taken from the appendix in Himmelblau (1972).

The iterative procedure takes the parameter values supplied by the grid search as starting values (unless starting values are supplied by the user). The program then updates the vector of parameter estimates by the Davidon-Fletcher-Powell method until either of the following occurs:

a) The convergence criterion is satisfied. The convergence criterion is set in the start-up file, FRONT2.000, by the parameter, TOL. Presently it is set such that, if the proportional change in

the likelihood function and each of the parameters is less than 0.00001, then the iterative procedure terminates.

b) The maximum number of iterations permitted is completed. This is presently set in FRONT2.000 to 100.

Both of these parameters may be altered by the user.

3.3 PROGRAM OUTPUT

The ordinary least-squares estimates, the estimates after the grid search and the final maximum-likelihood estimates are all presented in the output file. Approximate standard errors are taken from the direction matrix used in the final iteration of the iterative procedure. This approximation of the covariance matrix is also listed in the output. If the user is not satisfied with this matrix, then an alternative covariance matrix, suggested by Berndt, et al (1974), is calculated. The program uses this alternative covariance matrix estimator if the IBHHH variable in the start-up file, FRONT2.000, is set to 1.

Estimates of mean and individual technical efficiencies are calculated according to the expressions presented in Battese and Coelli (1991). An estimate of technical efficiency is calculated for each firm in each year. An estimate of the mean technical efficiency of all firms is also be presented for each year. If η is restricted to be zero then only single estimates of the technical efficiencies of each firm and a single estimate of mean technical efficiency is calculated, because technical efficiency is assumed to be time-invariant.

4. A SHORT EXAMPLE

To illustrate the use of FRONTIER Version 2.0, a data set has been generated and saved in the file EG1.DTA. Assumptions about the true values of the parameters are as follows:

N = 15

T = 5

total number of observations = 68 (7 discarded arbitrarily)

k = 2

X_1 drawn from a uniform distribution on 1-10

X_2 drawn from a uniform distribution on 1-100

$\beta_0 = 2.0$

$\beta_1 = 0.3$

$\beta_2 = 0.5$

$\sigma_v^2 = 0.04$

$\mu = 0.5$

$\sigma^2 = 0.09$

$\eta = 0.1$.

The values of σ_v^2 and σ^2 imply that:

$\sigma_s^2 = \sigma_v^2 + \sigma^2 = 0.04 + 0.09 = 0.13$

$\gamma = \sigma^2/\sigma_s^2 = 0.09/0.13 = 0.69$.

The data file, EG1.DTA, is listed in Table 1. Note that the columns are: firm number, time period, Y_{it}, X_{1it} and X_{2it}, from left to right. We prefer to use FRONTIER in batch mode rather than interactively. Hence, the instruction file, EG1.INS, was constructed by making a copy of the file, BLANK.INS, and inserting the relevant information using a text editor. The instruction file, EG1.INS, is listed in Table 2. The first 9 lines of the file are self explanatory through the comments on the right-hand side of the file.[1] The names of the data, instruction and output files cannot be any longer than 12 characters each. Note that on line 7 of the instruction file it is indicated that μ is to be restricted to zero.

[1] It should be mentioned that the comments in BLANK.INS and FRONT2.000 are not read by FRONTIER and hence users may have instruction files which are made from scratch with a text editor and contain no comments. This is not recommended, however, as it would be too easy to lose track of which input value belongs on which line.

This has been done because we had earlier estimated 4 forms of the model and obtained the following log-likelihood values:

Model	Restrictions	Log-likelihood value
1	$\mu=0$, $\eta=0$	-12.45
2	$\eta=0$	-12.03
3	$\mu=0$	-8.26
4	no restrictions	-8.03
OLS	$\mu=0$, $\eta=0$, $\gamma=0$	-29.18

Using generalized likelihood-ratio tests with a 5% level of significance we found that model (3), which had μ restricted to be zero, was the prefered model. This chi-square test is always prefered to an asymptotic t-test because the estimated standard errors are only approximate.

The output file, EG1.OUT, resulting from this instruction file having been sent to FRONTIER, is listed in Table 3. The values of the estimated coefficients may be compared to the values which were used to generate the sample. This process can obviously be repeated for a number of randomly generated samples to form a monte carlo study. While generating this particular data set the generated values of U_i and e^{-U_i} were printed. These are listed in Table 4 and may be compared with the technical efficiency estimates obtained for the final year of the sample (year 5).

Table 1 - Data File (eg1.dta)

1	1	14.96840	9.416352	35.13364
2	1	16.19860	4.643334	77.29733
3	1	9.185659	5.094714	89.79895
5	1	5.651365	8.716519	27.87824
6	1	8.155598	1.066191	92.17432
7	1	6.652379	0.2580738	97.90739
8	1	18.07955	6.333816	82.08398
9	1	2.789212	2.350246	38.87622
10	1	11.67540	1.076388	81.76085
11	1	3.565924	3.431948	9.475523
12	1	13.99463	4.032543	55.09601
13	1	20.61785	7.975470	73.12965
15	1	6.025520	0.3438121	65.38025
1	2	12.70686	2.440149	63.83928
2	2	23.87556	7.890738	59.24076
3	2	6.965972	2.906033	72.57392
4	2	9.319360	2.668065	68.91608
5	2	9.426871	4.219816	57.42408
6	2	17.91515	2.661315	87.84301
7	2	11.26414	2.454655	30.78941
8	2	12.17137	2.827333	93.73409
9	2	2.140435	0.4387420	35.96098
12	2	27.90314	6.751591	80.27460
13	2	10.71752	4.424754	49.88585
14	2	5.042543	1.583060	22.07209
15	2	6.010864	0.9072119	38.72705
1	3	3.702737	6.148555	5.322397
2	3	2.353757	0.4793847	2.520496
3	3	5.597053	1.954877	41.54480
4	3	9.962295	8.169494	68.38918
5	3	14.37167	4.054660	77.55609
6	3	21.57833	5.028795	77.81173
7	3	25.25393	6.937140	98.90436
8	3	13.87632	8.405021	42.73973
9	3	4.763233	2.502908	59.74079
10	3	9.226517	6.589737	18.08550
11	3	9.517958	7.148683	26.65066
12	3	12.59230	8.039665	50.48774
13	3	17.64803	4.896381	88.18217
14	3	14.56668	6.721718	30.45071
15	3	19.92185	4.195113	95.83395
1	4	10.04695	4.550846	36.70369
2	4	38.97824	7.223219	89.31245
4	4	9.570093	4.870962	50.01818
5	4	9.430373	9.311944	40.99634
6	4	12.93405	2.894831	63.05118
7	4	17.50535	8.085479	60.99176
8	4	19.63722	8.656345	94.15947
9	4	6.765616	3.426778	39.31178
10	4	15.19294	1.918254	78.62762

Table 1 - Data File (eg1.dta) continued

13	4	21.57045	9.328890	87.12386
14	4	11.56367	7.833866	60.33981
15	4	17.43632	5.620628	44.21768
1	5	10.69106	6.299923	30.67629
2	5	28.72547	2.469733	80.47305
4	5	17.41430	8.142350	37.83538
5	5	10.42365	5.941750	56.21796
6	5	15.35249	9.794919	14.92596
7	5	16.91270	3.668337	53.38867
8	5	11.38560	2.832359	67.88791
9	5	7.080967	1.498327	85.82528
10	5	13.22307	6.212032	28.16412
12	5	20.99534	2.261493	99.71692
13	5	17.75888	6.727894	76.70413
14	5	22.21073	7.720555	95.88951
15	5	17.82052	4.216369	41.68804
11	4	11.05639	6.177133	64.37679
12	4	1.556997	7.188464	1.073098

Table 2 - Instruction File (eg1.ins)

```
eg1.dta          DATA FILE NAME
eg1.out          OUTPUT FILE NAME
15               NUMBER OF FIRMS
5                NUMBER OF TIME PERIODS
68               TOTAL NUMBER OF OBSERVATIONS
2                NUMBER OF REGRESSOR VARIABLES
N                MU (Y/N)
Y                ETA (Y/N)
N                STARTING VALUES (Y/N)
                 IF YES THEN      B0
                                  B1 TO
                                  BK
                                  SIGMA SQUARED
                                  GAMMA
                                  MU
                                  ETA
```

Table 3 - Output File (eg1.out)

```
OUTPUT FROM THE PROGRAM FRONTIER (version 2.0)

THE OLS ESTIMATES ARE :

                     COEFFICIENT  STANDARD-ERROR         T-RATIO

  INTERCEPT      -0.10299895E+00  0.23648802E+00 -0.43553557E+00
  X 1             0.37353653E+00  0.55941207E-01  0.66773056E+01
  X 2             0.51800965E+00  0.56523434E-01  0.91645113E+01
  SIGMA-SQUARED   0.14451374E+00

LOG LIKELIHOOD FUNCTION =  -0.29184779E+02

THE ESTIMATES AFTER THE GRID SEARCH WERE :

  INTERCEPT       0.31720802E+00
  X 1             0.37353653E+00
  X 2             0.51800965E+00
  SIGMA-SQUARED   0.32630777E+00
  GAMMA           0.85000000E+00
  MU IS RESTRICTED TO BE ZERO
  ETA             0.50000000E-01

THE FINAL MLE ESTIMATES ARE :

                     COEFFICIENT  STANDARD-ERROR         T-RATIO

  INTERCEPT       0.32758622E+00  0.16321176E+00  0.20071238E+01
  X 1             0.34074197E+00  0.38708264E-01  0.88028225E+01
  X 2             0.53242275E+00  0.36430275E-01  0.14614843E+02
  SIGMA-SQUARED   0.19081528E+00  0.71495156E-01  0.26689259E+01
  GAMMA           0.77760260E+00  0.72916930E-01  0.10664226E+02
  MU IS RESTRICTED TO BE ZERO
  ETA             0.11394257E+00  0.44512095E-01  0.25598115E+01

LOG LIKELIHOOD FUNCTION =  -0.82557004E+01

VALUE OF CHI-SQUARE TEST OF ONE-SIDED ERROR =   0.41858158E+02
WITH DEGREES OF FREEDOM = 2

NUMBER OF ITERATIONS =  35

(MAXIMUM NUMBER OF ITERATIONS SET AT :  100)

NUMBER OF FIRMS =  15

NUMBER OF PERIODS =   5

TOTAL NUMBER OF OBSERVATIONS =  68

THUS THERE ARE:   7  OBSNS NOT IN THE PANEL
```

Table 3 - Output File (eg1.out) continued

```
COVARIANCE MATRIX :

  0.26638080E-01 -0.17313582E-02 -0.47332317E-02  0.41787345E-02
  0.35773371E-02 -0.10175208E-02
 -0.17313582E-02  0.14983297E-02 -0.10848577E-04  0.34952280E-03
  0.51860871E-03 -0.61110319E-03
 -0.47332317E-02 -0.10848577E-04  0.13271650E-02  0.87685684E-04
  0.23217409E-03  0.14696762E-04
  0.41787345E-02  0.34952280E-03  0.87685684E-04  0.51115574E-02
  0.45745094E-02 -0.10889522E-02
  0.35773371E-02  0.51860871E-03  0.23217409E-03  0.45745094E-02
  0.53168786E-02 -0.13494765E-02
 -0.10175208E-02 -0.61110319E-03  0.14696762E-04 -0.10889522E-02
 -0.13494765E-02  0.19813266E-02


TECHNICAL EFFICIENCY ESTIMATES FOR YEAR    1 :

 FIRM-NO.          TECH.-EFF.-EST.

       1           0.64171075E+00
       2           0.93719961E+00
       3           0.35867279E+00
       4     NO OBSERVATION IN THIS PERIOD
       5           0.40155731E+00
       6           0.74762573E+00
       7           0.77161616E+00
       8           0.57133789E+00
       9           0.25591661E+00
      10           0.77678745E+00
      11           0.49469205E+00
      12           0.71318020E+00
      13           0.62810524E+00
      14     NO OBSERVATION IN THIS PERIOD
      15           0.75193711E+00


 MEAN TECHNICAL EFF.=  0.65362018E+00
```

Table 3 - Output File (eg1.out) continued

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 2 :

FIRM-NO.	TECH.-EFF.-EST.
1	0.67271438E+00
2	0.94363894E+00
3	0.40022609E+00
4	0.49190378E+00
5	0.44275528E+00
6	0.77097449E+00
7	0.79302683E+00
8	0.60648055E+00
9	0.29619948E+00
10	NO OBSERVATION IN THIS PERIOD
11	NO OBSERVATION IN THIS PERIOD
12	0.73918814E+00
13	0.65997275E+00
14	0.62324284E+00
15	0.77494257E+00

MEAN TECHNICAL EFF.= 0.68074659E+00

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 3 :

FIRM-NO.	TECH.-EFF.-EST.
1	0.70172607E+00
2	0.94944828E+00
3	0.44142449E+00
4	0.53060858E+00
5	0.48313044E+00
6	0.79251547E+00
7	0.81272372E+00
8	0.63973667E+00
9	0.33750914E+00
10	0.81689490E+00
11	0.57011337E+00
12	0.76328562E+00
13	0.68985400E+00
14	0.65536523E+00
15	0.79615584E+00

MEAN TECHNICAL EFF.= 0.70656806E+00

Table 3 - Output File (eg1.out) continued

```
TECHNICAL EFFICIENCY ESTIMATES FOR YEAR    4 :

 FIRM-NO.           TECH.-EFF.-EST.

       1           0.72873798E+00
       2           0.95468323E+00
       3     NO OBSERVATION IN THIS PERIOD
       4           0.56778587E+00
       5           0.52230613E+00
       6           0.81231923E+00
       7           0.83078618E+00
       8           0.67101232E+00
       9           0.37924706E+00
      10           0.83452688E+00
      11           0.60533093E+00
      12           0.78552479E+00
      13           0.71772647E+00
      14           0.68551600E+00
      15           0.81564927E+00


 MEAN TECHNICAL EFF.=  0.73100413E+00



TECHNICAL EFFICIENCY ESTIMATES FOR YEAR    5 :

 FIRM-NO.           TECH.-EFF.-EST.

       1           0.75377508E+00
       2           0.95939573E+00
       3     NO OBSERVATION IN THIS PERIOD
       4           0.60321806E+00
       5           0.55997999E+00
       6           0.83046877E+00
       7           0.84730235E+00
       8           0.70026197E+00
       9           0.42086124E+00
      10           0.85065441E+00
      11     NO OBSERVATION IN THIS PERIOD
      12           0.80597622E+00
      13           0.74360346E+00
      14           0.71366562E+00
      15           0.83350701E+00


 MEAN TECHNICAL EFF.=  0.75400412E+00
```

Table 3 - Output File (eg1.out) continued

SUMMARY OF PANEL OF OBSERVATIONS:
(1=OBSERVED, 0=UNOBSERVED)

T:	1	2	3	4	5	
N						
1	1	1	1	1	1	5
2	1	1	1	1	1	5
3	1	1	1	0	0	3
4	0	1	1	1	1	4
5	1	1	1	1	1	5
6	1	1	1	1	1	5
7	1	1	1	1	1	5
8	1	1	1	1	1	5
9	1	1	1	1	1	5
10	1	0	1	1	1	4
11	1	0	1	1	0	3
12	1	1	1	1	1	5
13	1	1	1	1	1	5
14	0	1	1	1	1	4
15	1	1	1	1	1	5
	13	13	15	14	13	68

Table 4 - Generated Values of U_i and $TE_i = e^{-U_i}$

!_PRINT N U TE

N	U	TE
1.000000	0.3498407	0.7048004
2.000000	0.8741417E-01	0.9162975
3.000000	0.7738180	0.4612487
4.000000	0.7221281	0.4857175
5.000000	0.5870125	0.5559858
6.000000	0.3295963	0.7192140
7.000000	0.3006275	0.7403535
8.000000	0.4844381	0.6160433
9.000000	0.9888391	0.3720083
10.00000	0.2690496	0.7641053
11.00000	0.6402844	0.5271425
12.00000	0.4077790	0.6651259
13.00000	0.4766907	0.6208345
14.00000	0.4983612	0.6075254
15.00000	0.4865283	0.6147570

5. SOME FINAL POINTS

The original version of this program was written in 1985, see Coelli (1985). It was updated in 1986 to include the calculation of individual firm technical efficiencies, see Battese and Coelli (1988). The analysis of an unbalanced data panel required that the program be further altered in 1988, see Battese, Coelli and Colby (1989). This third revision provided the first version of FRONTIER to be used outside the University of New England. A number of copies of FRONTIER and the associated documentation (Coelli (1989)) were provided to interested researchers. Through this exposure a number of small bugs were found and reported back to the author. Many suggestions for improvement and extension of FRONTIER were also provided. As the program required revision to permit the specification of time-varying technical efficiency for Battese and Coelli (1991), we have addressed many of the suggestions of users and made significant changes to the program, the result being FRONTIER Version 2.0.

This version has been extensively tested using a number of monte carlo simulations but undoubtedly a few bugs remain. If users discover a bug, then it would be appreciated if they contact the author either by post or on computer mail:

tcoelli@gara.une.oz.au

Any users who have not been supplied a copy of the program directly by the author and who wish to be notified of any bugs or new versions, are advised to contact the author so that they may be put on the mailing list.

REFERENCES

Aigner, D.J., C.A.K. Lovell and P. Schmidt (1977), "Formulation and Estimation of Stochastic Frontier Production Function Models", *Journal of Econometrics*, 6, 21-37.

Battese, G.E. (1991), *Frontier Production Functions and Technical Efficiency: A Survey of Empirical Applications in Agricultural Economics*, Working Papers in Econometrics and Applied Statistics, No. 50, Department of Econometrics, University of New England, Armidale, pp.32.

Battese, G.E. and T.J. Coelli (1988), "Prediction of Firm-Level Technical Efficiencies With a Generalized Frontier Production Function and Panel Data", *Journal of Econometrics*, 38, 387-399.

Battese, G.E. and T.J. Coelli (1991), *Frontier Production Functions, Technical Efficiency and Panel Data: With Application to Paddy Farmers in India*, Working Papers in Econometrics and Applied Statistics, No. 56, Department of Econometrics, University of New England, Armidale.

Battese, G.E., T.J. Coelli and T.C. Colby (1989), "Estimation of Frontier Production Functions and the Efficiencies of Indian Farms Using Panel Data From ICRISAT's Village Level Studies", *Journal of Quantitative Economics*, 5, 327-348.

Battese, G.E. and G.S. Corra (1977), "Estimation of a Production Frontier Model: With Application to the Pastoral Zone of Eastern Australia", *Australian Journal of Agricultural Economics*, 21, 169-179.

Bauer, P.W. (1990), "Recent Developments in the Econometric Estimation of Frontiers", *Journal of Econometrics*, 46, 39-56.

Beck, M. (1991), *Empirical Applications of Frontier Functions: A Bibliography*, mimeo, Joachim-Ringelnatz-STR.20, W-6200 Wiesbaden, Germany, pp.9.

Berndt, E.K., B.H. Hall, R.E. Hall and J.A. Housman (1974), "Estimation and Inference in Non-Linear Structural Models", *Annals of Economic and Social Measurement*, 3, 653-665.

Coelli, T.J. (1985), *A generalised Frontier Production Function for Cross-Sectional, Time-Series Data*, Unpublished B.App.Ec. (Honours) Dissertation, University of New England, Armidale, NSW, Australia.

Coelli, T.J. (1989), *Estimation of Frontier Production Functions: A Guide to the Computer Program, FRONTIER*, Working Papers in Econometrics and Applied Statistics, No.34, Department of Econometrics, University of New England, Armidale, pp.31.

Forsund, F.R., C.A.K. Lovell and P. Schmidt (1980), "A Survey of Frontier Production Functions and of their Relationship to Efficiency Measurement", *Journal of Econometrics*, 13, 5-25.

Himmelblau, D.M. (1972), *Applied Non-Linear Programming*, McGraw-Hill, New York.

Hughes, M.D. (1988), "A Stochastic Frontier Cost Function for Residential Child Care Provision", *Journal of Applied Econometrics*, 3, 203-214.

Meeusen, W. and J. van den Broeck (1977), "Efficiency Estimation from Cobb-Douglas Production Functions With Composed Error", *International Economic Review*, 18, 435-444.

Ley, E. (1990), *A Bibliography on Production and Efficiency*, mimeo, Department of Economics, University of Ann Arbor, MI 48109, pp.32.

Pitt, M.M. and L.-F. Lee (1981), "Measurement and Sources of Technical Inefficiency in the Indonesian Weaving Industry", *Journal of Development Economics*, 9, 43-64.

Schmidt, P. (1986), "Frontier Production Functions", *Econometric Reviews*, 4, 289-328.

Stevenson, R.E. (1980), "Likelihood Functions for Generalized Stochastic Frontier Estimation", *Journal of Econometrics*, 13, 57-66.

APPENDIX A - PROGRAMMER'S GUIDE

A.1 The FRONT2.000 file

The start-up file, FRONT2.000, is listed in Table A1. Eleven values may be altered in FRONT2.000. A brief description of each follows.

1) IPRINT - If this is set to 1, then the program prints out the log-likelihood function value and the vector of parameter estimates at each iteration. The printing of information at each iteration is suppressed if IPRINT is set to 0.

2) INDIC - This relates to the Coggin uni-dimensional search which is conducted before each iteration to determine the optimal step length. It may be used as follows:

indic=2 : do not scale step length in uni-dimensional search;
indic=1 : scale (to length of last step) only if last step was smaller; and
indic=any other number : scale (to length of last step).

For more information see Himmelblau (1972).

3) TOL - This variable sets the convergence tolerance on the iterative process. If this value is say set to 0.00001 then the iterative procedure terminates when the proportional change in the log-likelihood function and in each of the estimated parameters are all less than 0.00001.

4) TOL2 - This variable sets the required tolerance on the Coggin uni-dimensional search done each iteration to determine the step length. For more information see Himmelblau (1972).

Table A1 - The start-up file FRONT2.000

```
KEY VALUES USED IN FRONTIER PROGRAM (VERSION 2.0)
NUMBER:         DESCRIPTION:
1           IPRINT - 1=PRINT ITERATIONS, 0=DO NOT
1           INDIC - USED IN UNIDIMENSIONAL SEARCH - SEE BELOW
0.00001     TOL - CONVERGENCE TOLERANCE (PROPORTIONAL)
0.001       TOL2 - TOLERANCE USED IN UNI-DIMENSIONAL SEARCH
1.0D+15     BIGNUM - USED TO SET BOUNDS ON DEN & DIST
0.0001      STEP1 - SIZE OF 1ST STEP IN SEARCH PROCEDURE
1           IGRID2 - 1=DOUBLE ACCURACY GRID SEARCH, 0=SINGLE
0.1         GRIDNO - STEPS TAKEN IN 1ST PHASE GRID SEARCH ON GAMMA
1.0         GWIDTH - WIDTH EACH SIDE OF 0 OF GRID SEARCH ON ETA
0           IBHHH - 1=BERNT HALL HALL & HOUSMAN S.E. ESTIMATES USED
100         MAXIT - MAXIMUM NUMBER OF ITERATIONS PERMITTED

THE NUMBERS IN THIS FILE ARE READ BY THE FRONTIER PROGRAM WHEN IT
BEGINS EXECUTION. YOU MAY CHANGE THE NUMBERS IN THIS FILE IF YOU
WISH. IT IS ADVISED THAT A BACKUP OF THIS FILE IS MADE PRIOR TO
ALTERATION.
```

5) BIGNUM - This variable is used to specicy largest (and implicitly the smallest) number that the program should deal with. Its primary use is to place bounds upon what the smallest number can be in the subroutines DEN and DIS which evaluate the standard normal density and distribution functions, respectively. Errors with numerical underflows and overflows were the problems most frequently encountered by people attempting to install the first version of this program on various mainframe computers. This number has been set to 1.0e+15 for the IBM PC through trial and error. If you plan to mount this program on a mainframe computer it is advised that you consult computer support staff on the correct setting of this number. It generally would be safe to leave it as it is. However, greater precision can be gained if larger numbers are permitted.

6) STEP1 - This variable is used to set the size of the first step in the iterative process. This should be set carefully as too large a value may result in the program 'stepping' right out of the parameter space.

7) IGRID2 - This is a flag which if set to 1 will cause the grid search to complete a second phase grid search around the estimates obtained in the first phase of the grid search. If set to 0, only the first phase of the grid search will be conducted. For more information refer to the description of the grid search in Section 3.

8) GRIDNO - This variable sets the width of the steps taken in the grid search between zero and one on γ. It will also indirectly influence the degree of precision in the grid search on μ and η. For more information refer to the description of the grid search in Section 3.

9) GWIDTH - This variable sets the width of the grid search on η around zero. That is, a value of 1.0 implies a search will be done from -1.0 to +1.0. The value of GWIDTH interacts with GRIDNO to determine how wide each step is in the grid search on η. Again, refer to the description of the grid search in Section 3 for more information.

10) IBHHH - This is a flag which allows the user to ask the that the covariance matrix estimator, proposed by Berndt, et al. (1974), be used instead of using the direction matrix from the last iteration of the Davidon-Fletcher-Powell algorithm to approximate the covariance matrix. In a few instances, the grid search may provide starting values so close to the minimum that only a small number of iterations are completed before convergence is achieved. This may mean that the direction matrix may not be a good estimator of the covariance matrix. Hence, the Berndt, et al. (1974) matrix may be preferred. If IBHHH is set to 1, this matrix will be used to calulate standard errors and t-ratios, while if set to 0, the final direction matrix will be used.

11) MAXIT - sets the maximum number of iterations that will be conducted. This is a handy option when large batch files are written for monte carlo simulation.

A.2 Subroutine Descriptions

The main subroutines in FRONTIER are organised as follows:

```
EXEC --- DATTA
     --- MINI --- GRID
                --- SEARCH
                --- ETA
                --- CONVG
     --- RESULT
```

with other subroutines:

FUN
DER
CHECK
OLS
INVERT

and functions:

DEN
DIS
ZI
EPE
EPR

A brief description of the purpose of each is now presented.

EXEC:
This is the main calling program. It firstly reads the start-up file, FRONT2.000, before calling DATTA to read in data, then MINI to find the maximum likelihood point, and lastly it calls RESULT to create the output.

SUBROUTINES

DATTA:
This subroutine reads instructions either from a file or from the terminal, then reads the data file and takes logarithms of the data.

MINI:
This is the main subroutine of the program. It firstly calls GRID to do the grid search (assuming starting values are not specified by the user). MINI then conducts the main iterative loop of FRONTIER, calling SEARCH, ETA and CONVRG repeatedly until the convergence criteria are satisfied (or the maximum number of iterations is achieved). The Davidon-Fletcher-Powell method is used.

RESULT:
Sends all final results to the output file. These include parameter estimates, approximate standard errors, t-ratios, and the individual and mean technical efficiency estimates.

GRID:
Does a grid search over γ, μ and η (if no restrictions imposed). The parameters β_0 and σ_s^2 are adjusted according to expectation theory.

SEARCH:
Performs a uni-dimensional search to determine the optimal step length of the next iteration. The Coggin method is used.

ETA:
This subroutine updates the direction matrix according to the Davidon-Fletcher-Powell method at each iteration. For more information, see Himmelblau (1972).

CONVRG:
Tests the convergence criterion at the end of each iteration. If the proportion change in the log-likelihood function and each of the parameters is no greater than the value of TOL (initially set to 0.00001) the iterative process will terminate.

FUN:
Calculates the negative of the log-likelihood function (LLF). Note that FRONTIER minimizes the negative of the LLF which is equivalent to maximizing the LLF.

DER:
Calculates the first partial derivatives of the negative of the LLF. This subroutine is also used to assist with the calculation of the Berndt, et al. (1974) covariance matrix.

CHECK:
Ensures that the estimated parameters do not venture outside the theoretical bounds (i.e. $0<\gamma<1$ and $\sigma_s^2>0$).

OLS:
Calculates the Ordinary Least-Squares estimates of the model to be used as starting values. It also calculates OLS standard errors which are presented in the final output.

INVERT:
Inverts a given matrix. It is called by OLS as well as by GRID to assist with the adjustment of σ_s^2 in the grid search.

FUNCTIONS

DEN:
Evaluates the density function of a standard normal distribution.

DIS:
Evaluates the distribution function of a standard normal distibution.

ZI:
Calculates the value of Z_i^* as defined in Battese and Coelli (1991).

EPE:
Calculates the product of the vector, η_i, and its transpose, as defined in Battese and Coelli (1991). EPE stands for 'Eta Prime Eta'.

EPR:
Calculates the product of the vector, η_i, and the transpose of the residual vector of the i-th firm. EPR stands for 'Eta Prime Residual'.

APPENDIX B - PROGRAM LISTING

```
      PROGRAM EXEC
C     This program uses the Davidon-Fletcher-Powell algorithm to
C     estimate using (unbalanced) panel data a frontier production
C     function which has a technical efficiency distribution with a
C     non-zero mode mu and a parameter eta to allow time-varying technical
C     efficiency (a model formulated by Battese and Coelli (1991)).
C     A large proportion of the SEARCH, CONVRG,MINI and ETA
C     subroutines are taken from Himmelblau (1972, appendix b).
C     The functions DIS and DENS are based upon subroutines in
C     Corra (1976,appendix B).
C     The remainder of the program is the work of Tim Coelli.
C     Any person is welcome to copy and use this program free of
C     charge. The author takes no responsibility for any
C     inconvenience caused by undetected errors. If an error is
C     detected the author would appreciate being informed.
C     A 1991 paper in the working paper series in the Department of
C     Econometrics, University of New England, Armidale, NSW 2351,
C     Australia describes the use of this program.
C     This version of FRONTIER is slightly different to earlier copies
C     distributed for mainframes. It has been altered to run on
C     an IBM PC/AT (and compatibles) with 640K using Lahey Fortran 77.
C     Last update = 27/May/1991
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
        COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      CHARACTER KOUTF*12
      OPEN(UNIT=70,FILE='front2.000')
      READ(70,*)
      READ(70,*)
      READ(70,*) IPRINT
      READ(70,*) INDIC
      READ(70,*) TOL
      READ(70,*) TOL2
      READ(70,*) BIGNUM
      READ(70,*) STEP1
      READ(70,*) IGRID2
      READ(70,*) GRIDNO
      READ(70,*) GWIDTH
      READ(70,*) IBHHH
      READ(70,*) MAXIT
      CLOSE(70)
      NFUNCT=0
      NDRV=0
      KOUTF=' '
      CALL DATTA(KOUTF)
      CALL MINI
      CALL RESULT(KOUTF)
      STOP
      END
```

```
      SUBROUTINE MINI
C     Contains the main loop of this iterative program.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/TWO/H(K4,K4),DELX(K4),DELG(K4),GX(K4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      COMMON/SEVEN/OB(K4),GB(K4),SV(K4),OBSE(K4)
      DIMENSION GY(K4)
      DO 98 I=1,K4
      GX(I)=0.0
      GY(I)=0.0
   98 CONTINUE
      IF (IGRID.EQ.1) THEN
      WRITE(6,*) 'CALCULATING OLS ESTIMATES...'
      WRITE(6,*) 'DOING GRID SEARCH...'
      CALL GRID
      ELSE
      DO 131 I=1,N
      SV(I)=Y(I)
      X(I)=Y(I)
  131 CONTINUE
      CALL FUN(X,FX)
      FY=FX
      END IF
      WRITE(6,*) 'START OF ITERATIVE PROCESS'
      ITER=0
      CALL DER(0,X,GX)
      WRITE(6,301) ITER,NFUNCT,-FY
      NC=1
  305 WRITE(6,302) (Y(I),I=NC,MIN(N,NC+4))
      NC=NC+5
      IF(NC.LE.N) GOTO 305
      IF (MAXIT.EQ.0) GOTO 70
    5 DO 20 I=1,N
      DO 10 J=1,N
   10 H(I,J)=0.0
   20 H(I,I)=1.0
      IF(IPRINT.EQ.1) WRITE(6,2100)
 2100     FORMAT(' GRADIENT STEP')
      DO 30 I=1,N
   30 S(I)=-GX(I)
   40 CALL SEARCH
      ITER=ITER+1
      IF (ITER.GE.MAXIT) THEN
      WRITE(6,*) 'MAXIMUM NUMBER OF ITERATIONS REACHED'
      GOTO 70
      ENDIF
    7 IF(FY.GT.FX) GOTO 5
      CALL DER(0,Y,GY)
      CALL CONVRG(IPASS)
      IF (IPASS.EQ.1.) GOTO 70
```

```
      IF(IPRINT.EQ.1) THEN
      WRITE(6,301) ITER,NFUNCT,-FY
      NC=1
  304 WRITE(6,302) (Y(I),I=NC,MIN(N,NC+4))
      NC=NC+5
      IF(NC.LE.N) GOTO 304
      ENDIF
      DO 50 I=1,N
      DELG(I)=GY(I)-GX(I)
      DELX(I)=Y(I)-X(I)
      GX(I)=GY(I)
   50 X(I)=Y(I)
      FX=FY
      CALL ETA
      DO 60 I=1,N
      S(I)=0.0
      DO 60 J=1,N
   60 S(I)=S(I)-H(I,J)*GY(J)
      GOTO 40
   70      CONTINUE
      WRITE(6,301)·ITER,NFUNCT,-FY
      NC=1
  303 WRITE(6,302) (Y(I),I=NC,MIN(N,NC+4))
      NC=NC+5
      IF(NC.LE.N) GOTO 303
  301 FORMAT(' ITERATION = ',I5,'  FUNC EVALS =',I7,'  LLF =',E16.8)
  302 FORMAT(4X,5E15.8)
      RETURN
      END

      SUBROUTINE CONVRG(IPASS)
C     Tests the convergence criterion.
C     The program is halted when the proportional change in the log-
C     likelihood and in each of the parameters is no greater than
C     a specified tolerance.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      XTOL=TOL
      FTOL=TOL
      IF(DABS(FX).LE.FTOL) GOTO 10
      IF(DABS((FX-FY)/FX).GT.FTOL) GOTO 60
      GOTO 20
   10 IF(DABS(FX-FY).GT.FTOL) GOTO 60
   20 DO 40 I=1,N
      IF(DABS(X(I)).LE.XTOL) GOTO 30
      IF(DABS((X(I)-Y(I))/X(I)).GT.XTOL) GOTO 60
      GOTO 40
   30    IF(DABS(X(I)-Y(I)).GT.XTOL) GOTO 60
   40    CONTINUE
      IPASS=1
      RETURN
```

```
 60     IPASS=2
        RETURN
        END

        SUBROUTINE ETA
C       Calculates the direction matrix (P).
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
        COMMON/TWO/H(K4,K4),DELX(K4),DELG(K4),GX(K4)
        COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
        COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
        DIMENSION HDG(K4),DGH(K4),HGX(K4)
        DXDG=0.0
        DGHDG=0.0
        DO 20 I=1,N
        HDG(I)=0.0
        DGH(I)=0.0
        DO 10 J=1,N
        HDG(I)=HDG(I)-H(I,J)*DELG(J)
 10     DGH(I)=DGH(I)+DELG(J)*H(J,I)
        DXDG=DXDG+DELX(I)*DELG(I)
 20     DGHDG=DGHDG+DGH(I)*DELG(I)
        DO 30 I=1,N
        DO 30 J=1,N
 30     H(I,J)=H(I,J)+DELX(I)*DELX(J)/DXDG+HDG(I)*DGH(J)/DGHDG
        DO 117 I=1,N
117     H(I,I)=DABS(H(I,I))
        DO 132 I=1,N
        HGX(I)=0.0
        DO 132 J=1,N
        HGX(I)=HGX(I)+H(I,J)*GX(J)
132     CONTINUE
        HGXX=0.
        GXX=0.
        DO 133 I=1,N
        HGXX=HGXX+HGX(I)**2
        GXX=GXX+GX(I)**2
133     CONTINUE
        C=0.
        DO 134 I=1,N
        C=C+HGX(I)*GX(I)
134     CONTINUE
        C=C/(HGXX*GXX)**0.5
        IF(DABS(C).LT.0.00001) THEN
        WRITE(6,*) 'ILL-CONDITIONED ETA'
        DO 136 I=1,N
        DO 137 J=1,N
137         H(I,J)=0.0
136         H(I,I)=DELX(I)/GX(I)
        ENDIF
        RETURN
        END

        SUBROUTINE SEARCH
```

```
C     UNIDIMENSIONAL SEARCH (COGGIN)
C     Determines the step length (t) using a unidimensional search.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/TWO/H(K4,K4),DELX(K4),DELG(K4),GX(K4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      IEXIT=0
      NTOL=0
      FTOL=TOL2
      FTOL2=FTOL/100.0
      FA=FX
      FB=FX
      FC=FX
      DA=0.0
      DB=0.0
      DC=0.0
      K=-2
      M=0
      STEP=STEP1
      D=STEP
      IF(INDIC.EQ.2.OR.ITER.EQ.0) GOTO 1
      DXNORM=0.0
      SNORM=0.0
      DO 102 I=1,N
      DXNORM=DXNORM+DELX(I)*DELX(I)
  102      SNORM=SNORM+S(I)*S(I)
      IF(INDIC.EQ.1.AND.DXNORM.GE.SNORM) GOTO 1
      RATIO=DXNORM/SNORM
      STEP=DSQRT(RATIO)
      D=STEP
    1 DO 2 I=1,N
    2 Y(I)=X(I)+D*S(I)
      CALL FUN(Y,F)
      K=K+1
      IF(F-FA) 5,3,6
    3 DO 4 I=1,N
    4 Y(I)=X(I)+DA*S(I)
      FY=FA
      IF(IPRINT.EQ.1) WRITE(6,2100)
 2100     FORMAT(' SEARCH FAILED. FN VAL INDEP OF SEARCH DIRECTION')
      GOTO 326
    5 FC=FB
      FB=FA
      FA=F
      DC=DB
      DB=DA
      DA=D
      D=2.0*D+STEP
      GOTO 1
    6 IF(K) 7,8,9
    7 FB=F
      DB=D
```

```
      D=-D
      STEP=-STEP
      GOTO 1
    8 FC=FB
      FB=FA
      FA=F
      DC=DB
      DB=DA
      DA=D
      GOTO 21
    9 DC=DB
      DB=DA
      DA=D
      FC=FB
      FB=FA
      FA=F
   10 D=0.5*(DA+DB)
      DO 11 I=1,N
   11 Y(I)=X(I)+D*S(I)
      CALL FUN(Y,F)
   12 IF((DC-D)*(D-DB)) 15,13,18
   13 DO 14 I=1,N
   14 Y(I)=X(I)+DB*S(I)
      FY=FB
      IF(IEXIT.EQ.1) GOTO 32
      IF(IPRINT.EQ.1) WRITE(6,2500)
 2500     FORMAT(' SEARCH FAILED. LOC OF MIN LIMITED BY ROUNDING')
      GOTO 325
   15 IF(F-FB) 16,13,17
   16 FC=FB
      FB=F
      DC=DB
      DB=D
      GOTO 21
   17 FA=F
      DA=D
      GOTO 21
   18 IF(F-FB) 19,13,20
   19 FA=FB
      FB=F
      DA=DB
      DB=D
      GOTO 21
   20 FC=F
      DC=D
   21 A=FA*(DB-DC)+FB*(DC-DA)+FC*(DA-DB)
      IF(A) 22,30,22
   22 D=0.5*((DB*DB-DC*DC)*FA+(DC*DC-DA*DA)*FB+(DA*DA-DB*DB)*FC)/A
      IF((DA-D)*(D-DC)) 13,13,23
   23 DO 24 I=1,N
   24 Y(I)=X(I)+D*S(I)
      CALL FUN(Y,F)
      IF(DABS(FB)-FTOL2) 25,25,26
   25 A=1.0
```

```
      GOTO 27
   26 A=1.0/FB
   27 IF((DABS(FB-F)*A)-FTOL) 28,28,12
   28 IEXIT=1
      IF(F-FB) 29,13,13
   29 FY=F
      GOTO 32
   30 IF(M) 31,31,13
   31 M=M+1
      GOTO 10
   32 DO 99 I=1,N
      IF(Y(I).NE.X(I)) GOTO 325
   99 CONTINUE
      GOTO 33
  325      IF(NTOL.NE.0.AND.IPRINT.EQ.1) WRITE(6,3000) NTOL
 3000      FORMAT(1X,'TOLERANCE REDUCED',I1,'TIME(S)')
  326      IF(FY.LT.FX) RETURN
      DO 101 I=1,N
      IF(S(I).NE.-GX(I)) RETURN
  101      CONTINUE
      WRITE(6,5000)
 5000      FORMAT(' SEARCH FAILED ON GRADIENT STEP, TERMINATION')
      RETURN
   33 IF(NTOL.EQ.5) GOTO 34
      IEXIT=0
      NTOL=NTOL+1
      FTOL=FTOL/10.
      GOTO 12
   34 IF(IPRINT.EQ.1) WRITE(6,2000)
 2000      FORMAT(' PT BETTER THAN ENTERING PT CANNOT BE FOUND')
      RETURN
      END

      SUBROUTINE CHECK
C     Checks if params are out of bounds & adjusts if required.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      N1=NB+1
      N2=NB+2
      IF(X(N1).LE.0.0) X(N1)=0.0001
      IF(X(N2).LE.0.01) X(N2)=0.01
      IF(X(N2).GE.0.99) X(N2)=0.99
      IF(Y(N1).LE.0.0) Y(N1)=0.0001
      IF(Y(N2).LE.0.01) Y(N2)=0.01
      IF(Y(N2).GE.0.99) Y(N2)=0.99
      RETURN
      END

      SUBROUTINE FUN(B,A)
C     Calculates the negative of the log-likelihood function.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      DATA PI/3.1415926/
      DIMENSION B(K4)
      CALL CHECK
      A=0.0
      F=DFLOAT(NN)
      FTOT=DFLOAT(NT)
      N1=NB+1
      N2=N1+1
      N3=N2+1
      A=0.5*FTOT*(DLOG(2.0*PI)+DLOG(B(N1)))
      A=A+0.5*(FTOT-F)*DLOG(1.0-B(N2))
      Z=B(N3)/(B(N1)*B(N2))**0.5
      A=A+F*DLOG(1.0-DIS(-Z))
      A=A+0.5*F*Z**2
      A2=0.0
      DO 132 I=1,NN
      A=A+0.5*DLOG(1.0+(EPE(I,B)-1.0)*B(N2))
      A=A-DLOG(1.0-DIS(-ZI(I,B)))
      DO 133 L=1,NP
      IF (XX(I,L,1).NE.0.0) THEN
      E=YY(I,L)
      DO 134 J=1,NB
      E=E-B(J)*XX(I,L,J)
  134 CONTINUE
      A2=A2+E**2
      END IF
  133 CONTINUE
      A=A-0.5*ZI(I,B)**2
  132 CONTINUE
      A=A+0.5*A2/((1.0-B(N2))*B(N1))
      NFUNCT=NFUNCT+1
      RETURN
      END

      SUBROUTINE DER(JJ,B,GX)
C     Calculates the fist-order partial derivatives of the negative
C     of the log-likelihood function and is also used to calculate
C     the i-th first-order partial derivative of the joint density
C     function of Y (used in the calculation of the covariance matrix
C     suggested by Berndt, Hall, Hall and Housman when the IBHHH flag is
C     set to 1 in the start-up file FRONT2.000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      DIMENSION B(K4),GX(K4)
      N1=NB+1
      N2=NB+2
      N3=NB+3
      N4=NB+4
      IF (NOMU.EQ.0) THEN
```

```
      N4=N3
      N3=NB+4
      END IF
      IF(JJ.EQ.0)THEN
      F=DFLOAT(NN)
      FTOT=DFLOAT(NT)
      KK=1
      MM=NN
      ELSE
      F=1.
      FTOT=DFLOAT(M(JJ))
      KK=JJ
      MM=JJ
      END IF
      FNP=DFLOAT(NP)

      Z=B(N3)/(B(N1)*B(N2))**0.5
      DO 132 J=1,NB
      A=0.0
      DO 133 I=KK,MM
      DO 134 L=1,NP
      E=YY(I,L)
      DO 135 K=1,NB
      E=E-XX(I,L,K)*B(K)
  135 CONTINUE
      A=A+XX(I,L,J)*E
  134 CONTINUE
  133 CONTINUE
      A=-A/(B(N1)*(1.-B(N2)))
      DO 136 I=KK,MM
      XPE=0.0
      DO 146 L=1,NP
      IF (XX(I,L,1).NE.0.0) THEN
      XPE=XPE+XX(I,L,J)*DEXP(-B(N4)*(DFLOAT(L)-FNP))
      END IF
   146     CONTINUE
      D=(DEN(-ZI(I,B))/(1.0-DIS(-ZI(I,B)))+ZI(I,B))*B(N2)*XPE
      A=A-D/(B(N2)*(1.0-B(N2))*B(N1)*(1.0+(EPE(I,B)-1.0)*B(N2)))**0.5
  136 CONTINUE
      GX(J)=A
  132 CONTINUE

      A=0.5*FTOT/B(N1)-0.5*F*(DEN(-Z)/(1.0-DIS(-Z))+Z)*Z/B(N1)
      SS=0.0
      DO 137 I=KK,MM
      A=A+.5*(DEN(-ZI(I,B))/(1.0-DIS(-ZI(I,B)))+ZI(I,B))*ZI(I,B)/B(N1)
      DO 138 L=1,NP
      E=YY(I,L)
      DO 139 J=1,NB
      E=E-XX(I,L,J)*B(J)
  139 CONTINUE
      SS=SS+E**2
  138 CONTINUE
  137 CONTINUE
```

```
      A=A-0.5*SS/((1.0-B(N2))*B(N1)**2)
      GX(N1)=A

      A=-0.5*(FTOT-F)/(1.0-B(N2))
      A=A-0.5*F*(DEN(-Z)/(1.0-DIS(-Z))+Z)*Z/B(N2)
      A=A+0.5*SS/((1.0-B(N2))**2*B(N1))
      DO 140 I=KK,MM
      A=A+0.5*(EPE(I,B)-1.0)/(1.0+(EPE(I,B)-1.0)*B(N2))
      D=B(N2)*(1.-B(N2))*(1.0+(EPE(I,B)-1.0)*B(N2))
      DZI=-(B(N3)+EPR(I,B))*D
      C=0.5*(B(N3)*(1.0-B(N2))-B(N2)*EPR(I,B))
      DZI=DZI-C*((1.0-2.0*B(N2))+(EPE(I,B)-1.0)*B(N2)*(2.0-3.0*B(N2)))
      DZI=DZI/(D**1.5*B(N1)**0.5)
      A=A-(DEN(-ZI(I,B))/(1.0-DIS(-ZI(I,B)))+ZI(I,B))*DZI
  140 CONTINUE
      GX(N2)=A

      IF (NOMU.EQ.1) THEN
      A=F/(B(N1)*B(N2))**0.5*(DEN(-Z)/(1.0-DIS(-Z))+Z)
      DO 142 I=KK,MM
      D=(DEN(-ZI(I,B))/(1.0-DIS(-ZI(I,B)))+ZI(I,B))*(1.-B(N2))
      A=A-D/(B(N2)*(1.-B(N2))*B(N1)*(1.+(EPE(I,B)-1.)*B(N2)))**.5
  142 CONTINUE
      GX(N3)=A
      END IF

      IF (NOETA.EQ.1) THEN
      A=0.0
      DO 151 I=KK,MM
      DE=0.0
      D=0.0
      DO 152 L=1,NP
      IF (XX(I,L,1).EQ.1) THEN
      T=DFLOAT(L)
      DE=DE-2.0*(T-1.0)*DEXP(-2.0*B(N4)*(T-FNP))
      E=YY(I,L)
      DO 153 J=1,NB
      E=E-XX(I,L,J)*B(J)
  153     CONTINUE
      D=D+(T-1.0)*DEXP(-B(N4)*(T-FNP))*E
      END IF
  152     CONTINUE
      DD=(B(N2)*(1.0-B(N2))*B(N1)*(1.0+(EPE(I,B)-1.0)*B(N2)))
      D=D*B(N2)*DD
      C=B(N3)*(1.0-B(N2))-B(N2)*EPR(I,B)
      C=C*0.5*B(N2)**2*(1.0-B(N2))*B(N1)*DE
      DZI=(D-C)/DD**1.5
      A=A-(DEN(-ZI(I,B))/(1.0-DIS(-ZI(I,B)))+ZI(I,B))*DZI
      A=A+B(N2)/2.0*DE/(1.0+(EPE(I,B)-1.0)*B(N2))
  151     CONTINUE
      GX(N4)=A
      END IF

      NDRV=NDRV+1
```

```
      RETURN
      END

      DOUBLE PRECISION FUNCTION DEN(A)
C     Evaluates the N(0,1) density function.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      DATA RRT2PI/ 0.3989422804/
      DEN=RRT2PI*DEXP(-0.5*A**2)
      IF (DEN.LT.1.0/BIGNUM) DEN=1.0/BIGNUM
      RETURN
      END

      DOUBLE PRECISION FUNCTION DIS(X)
C     Evaluates the N(0,1) distribution function.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      DIMENSION A(5),CONNOR(17)
      DATA CONNOR
     +  /8.0327350124D-17, 1.4483264644D-15, 2.4668270103D-14,
     +  3.9554295164D-13, 5.9477940136D-12, 8.3507027951D-11,
     +  1.0892221037D-9, 1.3122532964D-8, 1.4503852223D-7,
     +  1.4589169001D-6, 1.3227513228D-5, 1.0683760684D-4,
     +  7.5757575758D-4, 4.6296296296D-3, 2.3809523810D-2,
     +  0.1, 0.3333333333 /
      DATA RRT2PI/0.3989422804/
      S=X
      Y=S*S
      IF(S) 10,11,12
   11    P=0.5
      GOTO 31
   10    S=-S
   12    Z=RRT2PI*DEXP(-0.5*Y)
      IF(S-2.5) 13,14,14
   13    Y=-0.5*Y
      P=CONNOR(1)
      DO 15 L=2,17
   15    P=P*Y+CONNOR(L)
      P=(P*Y+1.0)*X*RRT2PI+0.5
      GOTO 31
   14    A(2)=1.0
      A(5)=1.0
      A(3)=1.0
      Y=1.0/Y
      A(4)=1.0+Y
      R=2.0
   19    DO 17 L=1,3,2
      DO 18 J=1,2
      K=L+J
      KA=7-K
   18    A(K)=A(KA)+A(K)*R*Y
   17    R=R+1.0
      IF(DABS(A(2)/A(3)-A(5)/A(4)).GT.1.0D-15) GOTO 19
   20    P=(A(5)/A(4))*Z/X
```

```
      IF(X) 21,11,22
   21    P=-P
      GOTO 31
   22    P=1.0-P
   31    CONTINUE
      IF(P.LT.1.0/BIGNUM) P=1.0/BIGNUM
      IF(P.GT.(1.0-1.0/BIGNUM)) P=1.0-1.0/BIGNUM
      DIS=P
      RETURN
      END

      DOUBLE PRECISION FUNCTION ZI(I,B)
C     Calculates the value of zi* as defined in Battese and Coelli (1991)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      DIMENSION B(K4)
      N1=NB+1
      N2=NB+2
      N3=NB+3
      A=B(N3)*(1.0-B(N2))-B(N2)*EPR(I,B)
      ZI=A/(B(N2)*(1.0-B(N2))*B(N1)*(1.0+(EPE(I,B)-1.0)*B(N2)))**0.5
      RETURN
      END

      DOUBLE PRECISION FUNCTION EPE(I,B)
C     Calculates eta prime eta.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      DIMENSION B(K4)
      IF (NOMU.EQ.1) THEN
      N4=NB+4
      ELSE
      N4=NB+3
      END IF
      TMP=0.0
      FNP=DFLOAT(NP)
      DO 101 L=1,NP
      IF (XX(I,L,1).NE.0.0) TMP=TMP+DEXP(-2.0*B(N4)*(DFLOAT(L)-FNP))
  101    CONTINUE
      EPE=TMP
      RETURN
      END

      DOUBLE PRECISION FUNCTION EPR(I,B)
C     Calculates eta prime residual.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      DIMENSION B(K4)
      IF (NOMU.EQ.1) THEN
      N4=NB+4
```

```
      ELSE
      N4=NB+3
      END IF
      TMP=0.0
      FNP=DFLOAT(NP)
      DO 101 L=1,NP
      IF (XX(I,L,1).NE.0.0) THEN
      E=YY(I,L)
      DO 102 J=1,NB
      E=E-B(J)*XX(I,L,J)
  102     CONTINUE
      TMP=TMP+E*DEXP(-B(N4)*(DFLOAT(L)-FNP))
      END IF
  101     CONTINUE
      EPR=TMP
      RETURN
      END

      SUBROUTINE DATTA(KOUTF)
C     Reads data in from a file (note that it is NOT in log-form).
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      CHARACTER KDATF*12,KINF*12,KOUTF*12,CHOP,CHMU,CHETA,CHST
      DIMENSION XXT(K3)
      NOMU=0
      NOETA=0
      CHMU='N'
      CHETA='N'
      IGRID=1
      DO 131 I=1,K4
      Y(I)=0.0
      X(I)=0.0
      S(I)=0.0
  131 CONTINUE
      WRITE(6,*) ' FRONTIER - Version 2.0'
      WRITE(6,*)'DO YOU WISH TO TYPE INSTRUCTIONS AT THE TERMINAL (T)'
      WRITE(6,*) '        OR USE AN INSTRUCTION FILE (F) ?'
      READ(5,61) CHOP
   61     FORMAT(A)
      IF ((CHOP.EQ.'T').OR.(CHOP.EQ.'t')) THEN
      WRITE(6,*) ' ENTER THE NAME OF YOUR DATA FILE :  '
      READ(5,60) KDATF
      WRITE(6,*) ' ENTER A NAME FOR AN OUTPUT FILE :  '
      READ(5,60) KOUTF
      WRITE(6,*) ' HOW MANY FIRMS IN THE DATA ?  '
      READ(5,*) NN
      IF(NN.GT.K1) THEN
      WRITE(6,70) K1
      STOP
      ENDIF
      WRITE(6,*) ' HOW MANY TIME PERIODS IN THE DATA ?  '
```

```
      READ(5,*) NP
      IF(NP.GT.K2) THEN
      WRITE(6,71) K2
      STOP
      ENDIF
      WRITE(6,*) ' HOW MANY OBSERVATIONS ARE THERE IN TOTAL ?  '
      READ(5,*) NT
      IF ((NN*NP).LT.NT) THEN
      WRITE(6,*) ' THE ABOVE NUMBER IS LARGER THAN THE PRODUCT OF THE'
      WRITE(6,*) '   PREVIOUS TWO ANSWERS - PROGRAM ABORT!'
      STOP
      END IF
      WRITE(6,*) ' HOW MANY REGRESSOR VARIABLES ARE THERE ?  '
      READ(5,*) NB
      IF(NB.GE.K3) THEN
      WRITE(6,72) K3
      STOP
      ENDIF
      WRITE(6,*) ' DOES THE MODEL INCLUDE MU ? (Y or N)  '
      READ(5,*) CHMU
      WRITE(6,*) ' DOES THE MODEL INCLUDE ETA ? (Y or N)  '
      READ(5,*) CHETA
      WRITE(6,*) ' IF YOU DO NOT WISH THE COMPUTER TO SELECT'
      WRITE(6,*) '   STARTING VALUES USING A GRID SEARCH YOU MUST'
      WRITE(6,*) '   SUPPLY THEM NOW'
      WRITE(6,*) ' DO YOU WISH TO SUPPLY STARTING VALUES ? (Y or N)  '
      READ(5,*) CHST
      IF ((CHST.EQ.'Y').OR.(CHST.EQ.'y')) THEN
      IGRID=0
   62     FORMAT(' ENTER STARTING VALUE FOR B',I2,' :  ')
      DO 138 I=1,NB+1
      WRITE(6,62) I-1
      READ(5,*) Y(I)
  138     CONTINUE
      WRITE(6,*) ' ENTER STARTING VALUE FOR SIGMA SQUARED :  '
      READ(5,*) Y(NB+2)
      WRITE(6,*) ' ENTER STARTING VALUE FOR GAMMA :  '
      READ(5,*) Y(NB+3)
      IF ((CHMU.EQ.'Y').OR.(CHMU.EQ.'y')) THEN
      WRITE(6,*) ' ENTER STARTING VALUE FOR MU :  '
      READ(5,*) Y(NB+4)
      IF ((CHETA.EQ.'Y').OR.(CHETA.EQ.'y')) THEN
      WRITE(6,*) ' ENTER STARTING VALUE FOR ETA :  '
      READ(5,*) Y(NB+5)
      END IF
      ELSE
      IF ((CHETA.EQ.'Y').OR.(CHETA.EQ.'y')) THEN
      WRITE(6,*) ' ENTER STARTING VALUE FOR ETA :  '
      READ(5,*) Y(NB+4)
      END IF
      END IF
      END IF
      ELSE IF ((CHOP.EQ.'F').OR.(CHOP.EQ.'f')) THEN
      WRITE(6,*) ' ENTER INSTRUCTION FILE NAME :  '
```

```
   READ(5,60) KINF
 60     FORMAT(A12)
   OPEN(UNIT=50,FILE=KINF)
   READ(50,60) KDATF
   READ(50,60) KOUTF
   READ(50,*) NN
   READ(50,*) NP
   READ(50,*) NT
   IF ((NN*NP).LT.NT) THEN
   WRITE(6,*) ' THE TOTAL NUMBER OF OBSNS EXCEEDS THE PRODUCT OF'
   WRITE(6,*) ' THE NUMBER OF FIRMS BY THE NUMBER OF YEARS - BYE!'
   STOP
   END IF
   READ(50,*) NB
   READ(50,61) CHMU
   READ(50,61) CHETA
   READ(50,61) CHST
   IF ((CHST.EQ.'Y').OR.(CHST.EQ.'y')) THEN
   IGRID=0
   DO 148 I=1,NB+3
   READ(50,*) Y(I)
148     CONTINUE
   IF ((CHMU.EQ.'Y').OR.(CHMU.EQ.'y')) THEN
   READ(50,*) Y(NB+4)
   IF ((CHETA.EQ.'Y').OR.(CHETA.EQ.'y')) READ(50,*) Y(NB+5)
   ELSE
   IF ((CHETA.EQ.'Y').OR.(CHETA.EQ.'y')) READ(50,*) Y(NB+4)
   END IF
   END IF
   ELSE
   WRITE(6,*) ' INCORRECT OPTION - MUST BE T OR F - BYE!'
   STOP
   ENDIF
70 FORMAT(' ERROR - MAXIMUM CROSS-SECTIONS ALLOWED = ',I4)
71 FORMAT(' ERROR - MAXIMUM TIME PERIODS ALLOWED = ',I4)
72 FORMAT(' ERROR - MAXIMUM REGRESSORS ALLOWED = ',I4)
   IF(NN.GT.K1) THEN
   WRITE(6,70) K1
   STOP
   ELSE IF(NP.GT.K2) THEN
   WRITE(6,71) K2
   STOP
   ELSE IF(NB.GE.K3) THEN
   WRITE(6,72) K3
   STOP
   ENDIF
   IF ((CHMU.EQ.'Y').OR.(CHMU.EQ.'y')) NOMU=1
   IF ((CHETA.EQ.'Y').OR.(CHETA.EQ.'y')) NOETA=1
   NB=NB+1
   N=NB+2+NOMU+NOETA
   OPEN(UNIT=40,FILE=KDATF)
   MINI=100
   MAXI=0
   MINT=20
```

```
      MAXT=0
      DO 132 I=1,NN
      M(I)=0
      DO 132 L=1,NP
      YY(I,L)=0.0
      DO 132 J=1,NB
      XX(I,L,J)=0.0
  132 CONTINUE
      DO 134 I=1,NT
      READ(40,*) FII,FTT,YYT,(XXT(J),J=2,NB)
      II=INT(FII)
      ITT=INT(FTT)
      IF (II.LT.MINI) MINI=II
      IF (II.GT.MAXI) MAXI=II
      IF (ITT.LT.MINT) MINT=ITT
      IF (ITT.GT.MAXT) MAXT=ITT
      M(II)=M(II)+1
      IF(YYT.LE.0.0) THEN
      WRITE(6,*) ' ZERO OR NEGATIVE NUMBER IN DATA - TERMINATION!'
      STOP
      ENDIF
      YY(II,ITT)=DLOG(YYT)
      DO 135 J=2,NB
      IF(XXT(J).LE.0.0) THEN
      WRITE(6,*) ' ZERO OR NEGATIVE NUMBER IN DATA - TERMINATION!'
      STOP
      ENDIF
      XX(II,ITT,J)=DLOG(XXT(J))
  135 CONTINUE
      XX(II,ITT,1)=1.0
  134 CONTINUE
      IF (MINI.LT.1) THEN
      WRITE(6,*) ' ERROR - A FIRM NUMBER IS < 1'
      STOP
      ELSE IF (MAXI.GT.NN) THEN
      WRITE(6,*) ' ERROR - A FIRM NUMBER IS > NUMBER OF FIRMS'
      STOP
      ELSE IF (MINT.LT.1) THEN
      WRITE(6,*) ' ERROR - A PERIOD NUMBER IS < 1'
      STOP
      ELSE IF (MAXT.GT.NP) THEN
      WRITE(6,*) ' ERROR - A PERIOD NUMBER IS > NUMBER OF PERIODS'
      STOP
      END IF
      DO 149 I=1,NN
      IF (M(I).EQ.0) THEN
      WRITE(6,66) I
   66     FORMAT(' ERROR - THERE ARE NO OBSERVATIONS ON FIRM ',I3)
      STOP
      END IF
  149     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RESULT(KOUTF)
C     Presents estimates, covariance matrix, standard errors and t-ratios, as
C     well as presenting many results including estimates of technical
C     efficiency.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/TWO/H(K4,K4),DELX(K4),DELG(K4),GX(K4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      COMMON/SEVEN/OB(K4),GB(K4),SV(K4),OBSE(K4)
      CHARACTER KOUTF*12
      DIMENSION COV(K4,K4),DX(K1,K4),CX(K4),MT(K2)
      DATA PI/3.1415926/
      OPEN (UNIT=70,FILE=KOUTF)
      N1=NB+1
      N2=NB+2
      N3=NB+3
      N4=NB+4
      IF (NOMU.EQ.0) THEN
      N4=N3
      N3=NB+4
      END IF
      WRITE (70,401)
  401 FORMAT(/,'OUTPUT FROM THE PROGRAM FRONTIER (Version 2.0)',//)
      IF (IGRID.EQ.0) THEN
      WRITE (70,402)
  402 FORMAT('THE STARTING VALUES SUPPLIED WERE : ',/)
  201 FORMAT('  INTERCEPT    ',E16.8)
  202 FORMAT('  X',I2,10X,E16.8)
  203 FORMAT('  SIGMA-SQUARED',E16.8)
  204 FORMAT('  GAMMA        ',E16.8)
  205 FORMAT('  MU           ',E16.8)
  206 FORMAT('  ETA          ',E16.8)
      WRITE (70,201) SV(1)
      DO 131 I=2,NB
      WRITE (70,202) I-1,SV(I)
  131 CONTINUE
      WRITE (70,203) SV(N1)
      WRITE (70,204) SV(N2)
      IF (NOMU.EQ.1) THEN
      WRITE (70,205) SV(N3)
      ELSE
      WRITE (70,*) '  MU IS RESTRICTED TO BE ZERO'
      END IF
      IF (NOETA.EQ.1) THEN
      WRITE (70,206) SV(N4)
      ELSE
      WRITE (70,*) '  ETA IS RESTRICTED TO BE ZERO'
      END IF
      ELSE
      WRITE (70,403)
  403 FORMAT(/,'THE OLS ESTIMATES ARE :',/)
```

```
      WRITE (70,404)
  404 FORMAT('                          COEFFICIENT  STANDARD-ERROR',
     +    '          T-RATIO',/)
      WRITE (70,301) OB(1),OBSE(1),OB(1)/OBSE(1)
      DO 132 I=2,NB
      WRITE (70,302) I-1,OB(I),OBSE(I),OB(I)/OBSE(I)
  132 CONTINUE
      WRITE (70,203) OB(N1)
      FNT=DFLOAT(NT)
      FNB=DFLOAT(NB)
      FXOLS=FNT/2.0*(DLOG(2.0*PI)+DLOG(OB(N1)*(FNT-FNB)/FNT)+1.0)
      WRITE(70,501) -FXOLS
      WRITE (70,405)
  405 FORMAT(/,'THE ESTIMATES AFTER THE GRID SEARCH WERE :',/)
      WRITE (70,201) GB(1)
      DO 133 I=2,NB
      WRITE (70,202) I-1,GB(I)
  133 CONTINUE
      WRITE (70,203) GB(N1)
      WRITE (70,204) GB(N2)
      IF (NOMU.EQ.1) THEN
      WRITE (70,205) GB(N3)
      ELSE
      WRITE (70,*) '  MU IS RESTRICTED TO BE ZERO'
      END IF
      IF (NOETA.EQ.1) THEN
      WRITE (70,206) GB(N4)
      ELSE
      WRITE (70,*) '  ETA IS RESTRICTED TO BE ZERO'
      END IF
      END IF
      IF(IBHHH.EQ.1) THEN
      DO 141 I=1,NN
      II=I
      CALL DER(II,Y,CX)
      DO 141 J=1,N
      DX(I,J)=-CX(J)
  141 CONTINUE
      DO 142 J=1,N
      DO 142 K=1,N
      COV(J,K)=0.0
      DO 142 I=1,NN
      COV(J,K)=COV(J,K)+DX(I,J)*DX(I,K)
  142 CONTINUE
      CALL INVERT(COV,N)
      DO 143 I=1,N
      DO 143 J=1,N
      H(I,J)=COV(I,J)
  143 CONTINUE
      END IF
      WRITE (70,406)
  406 FORMAT(//,'THE FINAL MLE ESTIMATES ARE :',/)
      WRITE (70,404)
  301 FORMAT('  INTERCEPT    ',3E16.8)
```

```
302 FORMAT('  X',I2,10X,3E16.8)
303 FORMAT('  SIGMA-SQUARED',3E16.8)
304 FORMAT('  GAMMA         ',3E16.8)
305 FORMAT('  MU            ',3E16.8)
306 FORMAT('  ETA           ',3E16.8)
    WRITE (70,301) Y(1),H(1,1)**0.5,Y(1)/H(1,1)**0.5
    DO 134 I=2,NB
    WRITE (70,302) I-1,Y(I),H(I,I)**0.5,Y(I)/H(I,I)**0.5
134 CONTINUE
    WRITE (70,303) Y(N1),H(N1,N1)**0.5,Y(N1)/H(N1,N1)**0.5
    WRITE (70,304) Y(N2),H(N2,N2)**0.5,Y(N2)/H(N2,N2)**0.5
    IF (NOMU.EQ.1) THEN
    WRITE (70,305) Y(N3),H(N3,N3)**0.5,Y(N3)/H(N3,N3)**0.5
    ELSE
    WRITE (70,*) '  MU IS RESTRICTED TO BE ZERO'
    END IF
    IF (NOETA.EQ.1) THEN
    WRITE (70,306) Y(N4),H(N4,N4)**0.5,Y(N4)/H(N4,N4)**0.5
    ELSE
    WRITE (70,*) '  ETA IS RESTRICTED TO BE ZERO'
    END IF
    IF (IBHHH.EQ.1) THEN
    WRITE(70,421)
421     FORMAT(/,'(NOTE: THE STANDARD ERRORS OF THE ABOVE MLEs CAME',
    + /,'       FROM THE BERNDT, HALL, HALL & HOUSMAN MATRIX)')
    ENDIF
    WRITE (70,501) -FX
501 FORMAT(/,'LOG LIKELIHOOD FUNCTION = ',E16.8)
    IF(IGRID.EQ.1) THEN
    CHI=2.0*DABS(FX-FXOLS)
    WRITE(70,5011) CHI
5011    FORMAT(/,'VALUE OF CHI-SQUARE TEST OF ONE-SIDED ERROR = ',E16.8)
    IDF=NOMU+NOETA+1
    WRITE(70,5012) IDF
5012    FORMAT('WITH DEGREES OF FREEDOM = ',I1)
    END IF
    WRITE (70,502) ITER
502 FORMAT(/,'NUMBER OF ITERATIONS = ', I3)
    WRITE(70,420) MAXIT
420 FORMAT(/,'(MAXIMUM NUMBER OF ITERATIONS SET AT :',I5,')')
    WRITE (70,511) NN
511 FORMAT(/,'NUMBER OF FIRMS = ',I3)
    WRITE(70,512) NP
512 FORMAT(/,'NUMBER OF PERIODS = ',I3)
    WRITE(70,513) NT
513 FORMAT(/,'TOTAL NUMBER OF OBSERVATIONS = ',I3)
    WRITE(70,514) NN*NP-NT
514 FORMAT(/,'THUS THERE ARE: ',I3,'  OBSNS NOT IN THE PANEL')
    WRITE (70,58)
 58 FORMAT(//,'COVARIANCE MATRIX :',/)
C   FIXFORMAT
 52 FORMAT(10(5E16.8,/))
    DO 135 I=1,N
    WRITE (70,52) (H(I,J),J=1,N)
```

```
 135 CONTINUE
     IF(IBHHH.EQ.1) THEN
     WRITE(70,*)
     WRITE(70,*) ' (NOTE: THIS IS THE BHHH COV MATRIX)'
     ENDIF

     IF (NOETA.EQ.0) WRITE(70,503)
 503 FORMAT(///,'TECHNICAL EFFICIENCY ESTIMATES :',/)
 504 FORMAT(//,'TECHNICAL EFFICIENCY ESTIMATES FOR YEAR ',I3,' :')
  68    FORMAT(/,' FIRM-NO.          TECH.-EFF.-EST.',/)
  69    FORMAT(5X,I3,9X,E16.8)
 505 FORMAT(5X,I3,4X,'NO OBSERVATION IN THIS PERIOD')
     FNP=DFLOAT(NP)
     DO 138 L=1,NP
     IF (NOETA.EQ.1) WRITE(70,504) L
     WRITE(70,68)
     T=DFLOAT(L)
     ETA=DEXP(-Y(N4)*(T-FNP))
     DO 136 I=1,NN
     IF ((XX(I,L,1).EQ.1.0).OR.(NOETA.EQ.0)) THEN
     FI=(Y(N3)*(1.0-Y(N2))-Y(N2)*EPR(I,Y))/(1.0+(EPE(I,Y)-1.0)*Y(N2))
     SI2=Y(N2)*(1.0-Y(N2))*Y(N1)/(1.0+(EPE(I,Y)-1.0)*Y(N2))
     SI=SI2**0.5
     TEI=(1.0-DIS(SI*ETA-FI/SI))/(1.0-DIS(-FI/SI))
     TEI=TEI*DEXP(-FI*ETA+0.5*SI2*ETA**2)
     WRITE(70,69) I,TEI
     ELSE
     WRITE(70,505) I
     END IF
 136 CONTINUE
     Z=Y(N3)/(Y(N1)*Y(N2))**0.5
     TE=(1.0-DIS(ETA*(Y(N2)*Y(N1))**0.5-Z))/(1.0-DIS(-Z))
     TE=TE*DEXP(-ETA*Y(N3)+0.5*Y(N2)*Y(N1)*ETA**2)
     WRITE(70,67) TE
  67    FORMAT(//,1X,'MEAN TECHNICAL EFF.=',E16.8,////)
     IF (NOETA.EQ.0) GOTO 71
 138 CONTINUE
  71 CONTINUE

     WRITE(70,441)
 441 FORMAT('SUMMARY OF PANEL OF OBSERVATIONS:',/,
     +     '(1=OBSERVED, 0=UNOBSERVED)',/)
     DO 449 L=1,NP
     MT(L)=L
 449 CONTINUE
     WRITE(70,442) (MT(L),L=1,NP)
C    FIXFORMAT
 442 FORMAT('  T:',100I4)
     WRITE(70,*) '  N'
C    FIXFORMAT
 443 FORMAT(102I4)
     DO 450 I=1,NN
     WRITE(70,443) I,(INT(XX(I,L,1)),L=1,NP),M(I)
 450 CONTINUE
```

```
      DO 451 L=1,NP
      MT(L)=0
      DO 451 I=1,NN
      MT(L)=MT(L)+INT(XX(I,L,1))
  451 CONTINUE
      WRITE(70,444) (MT(L),L=1,NP),NT
C     FIXFORMAT
  444 FORMAT(/,4X,101I4)
      WRITE(70,445)
  445 FORMAT(////)
      RETURN
      END


      SUBROUTINE GRID
C     Does a grid search
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/ONE/X(K4),Y(K4),S(K4),FX,FY
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      COMMON/SEVEN/OB(K4),GB(K4),SV(K4),OBSE(K4)
      DATA PI/3.1415926/
      DIMENSION A(K4,K4),B(K4,K4),C(K4,K4),XB(K1,K3)
      N1=NB+1
      N2=NB+2
      N3=NB+3
      N4=NB+4
      IF (NOMU.EQ.0) THEN
      N4=N3
      N3=NB+4
      END IF
      F=DFLOAT(NN)
      FNT=DFLOAT(NT)
      CALL OLS(OB,OBSE)
      DO 131 I=1,N
      Y(I)=OB(I)
  131 CONTINUE
      BO=Y(1)
      VAR=Y(N1)
      FX=BIGNUM
      DO 151 I=1,NN
      DO 151 J=1,NB
      XB(I,J)=0.0
      DO 152 L=1,NP
      XB(I,J)=XB(I,J)+XX(I,L,J)
   152     CONTINUE
      XB(I,J)=XB(I,J)/DFLOAT(M(I))
   151     CONTINUE
      DO 132 L=1,NB
      DO 132 K=1,NB
      A(L,K)=0.0
      B(L,K)=0.0
```

```
      C(L,K)=0.0
      DO 132 I=1,NN
      DO 132 J=1,NP
      A(L,K)=A(L,K)+XX(I,J,L)*XX(I,J,K)
  132 CONTINUE
      CALL INVERT(A,NB)
      DO 133 I=1,NN
      DO 133 L=1,NB
      DO 133 K=1,NB
      B(L,K)=B(L,K)+XB(I,L)*XB(I,K)
  133 CONTINUE
      TRACE=0.0
      DO 134 L=1,NB
      DO 135 K=1,NB
      DO 135 I=1,NB
      C(L,K)=C(L,K)+A(L,I)*B(I,K)
  135 CONTINUE
      TRACE=TRACE+C(L,L)
  134 CONTINUE
      R=(FNT-(FNT/F)**2*TRACE)/(FNT-DFLOAT(NB))
      DEV=0.0
      DEV2=0.0
      IF (NOMU.EQ.1) DEV=2.*Y(N1)**0.5
      IF (NOETA.EQ.1) DEV2=GWIDTH
      GY8=2.0*GWIDTH*GRIDNO
      GY7=2.0*2.0*Y(N1)**0.5*GRIDNO
      Y6B=GRIDNO
      Y6T=1.0-GRIDNO
      DO 136 Y8=-DEV2,DEV2,GY8
      Y(N4)=Y8
      DO 136 Y7=-DEV,DEV,GY7
      Y(N3)=Y7
      DO 137 Y6=Y6B,Y6T,GRIDNO
      Y(N2)=Y6
      Y(N1)=VAR/(1.+(R*((PI-2.)/PI)-1.)*Y(N2))
      Z=Y(N3)/(Y(N2)*Y(N1))**0.5
      Z=(Y(N2)*Y(N1))**0.5*DEN(-Z)/(1.-DIS(-Z))
      Y(1)=B0+Y(N3)+Z
      CALL FUN(Y,FY)
      IF(FY.LT.FX) THEN
      FX=FY
      DO 138 I=1,N
      X(I)=Y(I)
  138 CONTINUE
      END IF
  137 CONTINUE
  136 CONTINUE
      IF(IGRID2.EQ.1) THEN
      TEMP=GRIDNO/2.0
      TEMP2=GRIDNO/2.0
      IF (NOMU.EQ.0) TEMP=0.
      IF (NOETA.EQ.0) TEMP2=0.
      AA1=X(N3)-TEMP
      AA2=X(N3)+TEMP
```

```
      AA3=GRIDNO/5.0
      CC1=X(N4)-TEMP2
      CC2=X(N4)+TEMP2
      CC3=AA3
      BB1=X(N2)-GRIDNO/2.0
      BB2=X(N2)+GRIDNO/2.0
      BB3=AA3
      DO 139 Y8=CC1,CC2,CC3
      Y(N4)=Y8
      DO 139 Y7=AA1,AA2,AA3
      Y(N3)=Y7
      DO 140 Y6=BB1,BB2,BB3
      Y(N2)=Y6
      Y(N1)=VAR/(1.+(R*((PI-2.)/PI)-1.)*Y(N2))
      Z=Y(N3)/(Y(N2)*Y(N1))**0.5
      Z=(Y(N2)*Y(N1))**0.5*DEN(-Z)/(1.-DIS(-Z))
      Y(1)=B0+Y(N3)+Z
C     IF(Y(1).LT.(B0+DEV)) THEN
      CALL FUN(Y,FY)
      IF(FY.LT.FX) THEN
      FX=FY
      DO 141 I=1,N
      X(I)=Y(I)
  141 CONTINUE
      END IF
C     END IF
  140 CONTINUE
  139 CONTINUE
      END IF
      DO 142 I=1,N
      GB(I)=X(I)
  142 CONTINUE
      RETURN
      END



      SUBROUTINE INVERT(XX,N)
C     Finds the inverse of a given matrix.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GRIDNO,GWIDTH,IBHHH
      DIMENSION XX(K4,K4),IPIV(K4)
      DO 1 I=1,N
    1 IPIV(I)=0
      DO 11 I=1,N
      AMAX=0.
      DO 5 J=1,N
      IF(IPIV(J))2,2,5
    2 IF(DABS(XX(J,J))-AMAX) 4,4,3
    3 ICOL=J
      AMAX=DABS(XX(J,J))
    4 CONTINUE
    5 CONTINUE
```

```
      IPIV(ICOL)=1
      IF(AMAX-1.0/BIGNUM)6,6,7
    6 WRITE(6,*) 'SINGULAR MATRIX'
      STOP
    7 CONTINUE
      AMAX=XX(ICOL,ICOL)
      XX(ICOL,ICOL)=1.0
      DO 8 K=1,N
    8 XX(ICOL,K)=XX(ICOL,K)/AMAX
      DO 11 J=1,N
      IF(J-ICOL)9,11,9
    9 AMAX=XX(J,ICOL)
      XX(J,ICOL)=0.
      DO 10 K=1,N
   10 XX(J,K)=XX(J,K)-XX(ICOL,K)*AMAX
   11 CONTINUE
      RETURN
      END


      SUBROUTINE OLS(OB,OBSE)
C     Calculates the OLS estimates and their SEs.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(K1=100,K2=20,K3=16,K4=K3+4)
      COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
         COMMON/FR/M(K1),XX(K1,K2,K3),YY(K1,K2),NN,NT,NP,NOMU,NOETA,NB
      DIMENSION OB(K4),XPX(K4,K4),XPY(K4),OBSE(K4),MX(K4)
C     Calculate X'X and X'y
      DO 131 I=1,NB
      DO 132 J=1,NB
      XPX(I,J)=0.0
      DO 132 K=1,NN
      DO 132 L=1,NP
      XPX(I,J)=XPX(I,J)+XX(K,L,I)*XX(K,L,J)
  132 CONTINUE
      XPY(I)=0.0
      DO 131 K=1,NN
      DO 131 L=1,NP
      XPY(I)=XPY(I)+XX(K,L,I)*YY(K,L)
  131 CONTINUE
C     Determine correct scaling for X'X
      DO 120 I=1,NB
      H=(1.0-DLOG10(XPX(I,I)))/2.0
      IF (H.LT.0.0) GOTO 121
      MX(I)=H
      GOTO 120
  121      MX(I)=H-1
  120      CONTINUE
C     Scale, invert and then scale back
      IS=0
  123      IS=IS+1
      DO 122 I=1,NB
      DO 122 J=1,NB
      XPX(I,J)=XPX(I,J)*10.0**(MX(I)+MX(J))
```

```
  122     CONTINUE
      IF (IS.EQ.1) THEN
      CALL INVERT(XPX,NB)
      GOTO 123
      ENDIF
C     Calculate b=inv(X'X)X'y
      DO 133 I=1,NB
      OB(I)=0.0
      DO 133 J=1,NB
      OB(I)=OB(I)+XPX(I,J)*XPY(J)
  133 CONTINUE
      SS=0.0
      DO 134 K=1,NN
      DO 134 L=1,NP
      EE=YY(K,L)
      DO 135 I=1,NB
      EE=EE-XX(K,L,I)*OB(I)
  135 CONTINUE
      SS=SS+EE**2
  134 CONTINUE
      OB(NB+1)=SS/(NT-NB)
      DO 136 I=1,NB
      OBSE(I)=(OB(NB+1)*XPX(I,I))**0.5
  136 CONTINUE
      RETURN
      END
```

WORKING PAPERS IN ECONOMETRICS AND APPLIED STATISTICS

The Prior Likelihood and Best Linear Unbiased Prediction in Stochastic Coefficient Linear Models. Lung-Fei Lee and William E. Griffiths, No. 1 - March 1979.

Stability Conditions in the Use of Fixed Requirement Approach to Manpower Planning Models. Howard E. Doran and Rozany R. Deen, No. 2 - March 1979.

A Note on A Bayesian Estimator in an Autocorrelated Error Model. William Griffiths and Dan Dao, No. 3 - April 1979.

On R^2-Statistics for the General Linear Model with Nonscalar Covariance Matrix. G.E. Battese and W.E. Griffiths, No. 4 - April 1979.

Construction of Cost-Of-Living Index Numbers - A Unified Approach. D.S. Prasada Rao, No. 5 - April 1979.

Omission of the Weighted First Observation in an Autocorrelated Regression Model: A Discussion of Loss of Efficiency. Howard E. Doran, No. 6 - June 1979.

Estimation of Household Expenditure Functions: An Application of a Class of Heteroscedastic Regression Models. George E. Battese and Bruce P. Bonyhady, No. 7 - September 1979.

The Demand for Sawn Timber: An Application of the Diewert Cost Function. Howard E. Doran and David F. Williams, No. 8 - September 1979.

A New System of Log Change Index Numbers for Multilateral Comparisons. D.S. Prasada Rao, No. 9 - October 1980.

A Comparison of Purchasing Power Parity Between the Pound Sterling and the Australian Dollar - 1979. W.F. Shepherd and D.S. Prasada Rao, No. 10 - October 1980.

Using Time-Series and Cross-Section Data to Estimate a Production Function with Positive and Negative Marginal Risks. W.E. Griffiths and J.R. Anderson, No. 11 - December 1980.

A Lack-Of-Fit Test in the Presence of Heteroscedasticity. Howard E. Doran and Jan Kmenta, No. 12 - April 1981.

On the Relative Efficiency of Estimators Which Include the Initial Observations in the Estimation of Seemingly Unrelated Regressions with First Order Autoregressive Disturbances. H.E. Doran and W.E. Griffiths, No. 13 - June 1981.

An Analysis of the Linkages Between the Consumer Price Index and the Average Minimum Weekly Wage Rate. Pauline Beesley, No. 14 - July 1981.

An Error Components Model for Prediction of County Crop Areas Using Survey and Satellite Data. George E. Battese and Wayne A. Fuller, No. 15 - February 1982.

Networking or Transhipment? Optimisation Alternatives for Plant Location Decisions. H.I. Toft and P.A. Cassidy, No. 16 - February 1985.

Diagnostic Tests for the Partial Adjustment and Adaptive Expectations Models. H.E. Doran, No. 17 - February 1985.

A Further Consideration of Causal Relationships Between Wages and Prices. J.W.B. Guise and P.A.A. Beesley, No. 18 - February 1985.

A Monte Carlo Evaluation of the Power of Some Tests For Heteroscedasticity. W.E. Griffiths and K. Surekha, No. 19 - August 1985.

A Walrasian Exchange Equilibrium Interpretation of the Geary-Khamis International Prices. D.S. Prasada Rao, No. 20 - October 1985.

On Using Durbin's h-Test to Validate the Partial Adjustment Model. H.E. Doran, No. 21 - November 1985.

An Investigation into the Small Sample Properties of Covariance Matrix and Pre-Test Estimators for the Probit Model. William E. Griffiths, R. Carter Hill and Peter J. Pope, No. 22 - November 1985.

A Bayesian Framework for Optimal Input Allocation with an Uncertain Stochastic Production Function. William E. Griffiths, No. 23 - February 1986.

A Frontier Production Function for Panel Data: With Application to the Australian Dairy Industry. T.J. Coelli and G.E. Battese. No. 24 - February 1986.

Identification and Estimation of Elasticities of Substitution for Firm-Level Production Functions Using Aggregative Data. George E. Battese and Sohail J. Malik, No. 25 - April 1986.

Estimation of Elasticities of Substitution for CES Production Functions Using Aggregative Data on Selected Manufacturing Industries in Pakistan. George E. Battese and Sohail J. Malik, No. 26 - April 1986.

Estimation of Elasticities of Substitution for CES and VES Production Functions Using Firm-Level Data for Food-Processing Industries in Pakistan. George E. Battese and Sohail J. Malik, No. 27 - May 1986.

On the Prediction of Technical Efficiencies, Given the Specifications of a Generalized Frontier Production Function and Panel Data on Sample Firms. George E. Battese, No. 28 - June 1986.

A General Equilibrium Approach to the Construction of Multilateral Index Numbers. D.S. Prasada Rao and J. Salazar-Carrillo, No. 29 - August 1986.

Further Results on Interval Estimation in an AR(1) Error Model. H.E. Doran, W.E. Griffiths and P.A. Beesley, No. 30 - August 1987.

Bayesian Econometrics and How to Get Rid of Those Wrong Signs. William E. Griffiths, No. 31 - November 1987.

Confidence Intervals for the Expected Average Marginal Products of Cobb-Douglas Factors With Applications of Estimating Shadow Prices and Testing for Risk Aversion. Chris M. Alaouze, No. 32 - September, 1988.

Estimation of Frontier Production Functions and the Efficiencies of Indian Farms Using Panel Data from ICRISAT's Village Level Studies. G.E. Battese, T.J. Coelli and T.C. Colby, No. 33 - January, 1989.

Estimation of Frontier Production Functions: A Guide to the Computer Program, FRONTIER. Tim J. Coelli, No. 34 - February, 1989.

An Introduction to Australian Economy-Wide Modelling. Colin P. Hargreaves, No. 35 - February, 1989.

Testing and Estimating Location VEctors Under Heteroskedasticity. William Griffiths and George Judge, No. 36 - February, 1989.

The Management of Irrigation Water During Drought. Chris M. Alaouze, No. 37 - April, 1989.

An Additive Property of the Inverse of the Survivor Function and the Inverse of the Distribution Function of a Strictly Positive Random Variable with Applications to Water Allocation Problems. Chris M. Alaouze, No. 38 - July, 1989.

A Mixed Integer Linear Programming Evaluation of Salinity and Waterlogging Control Options in the Murray-Darling Basin of Australia. Chris M. Alaouze and Campbell R. Fitzpatrick, No. 39 - August, 1989.

Estimation of Risk Effects with Seemingly Unrelated Regressions and Panel Data. Guang H. Wan, William E. Griffiths and Jock R. Anderson, No. 40 - September 1989.

The Optimality of Capacity Sharing in Stochastic Dynamic Programming Problems of Shared Reservoir Operation. Chris M. Alaouze, No. 41 - November, 1989.

Confidence Intervals for Impulse Responses from VAR Models: A Comparison of Asymptotic Theory and Simulation Approaches. William Griffiths and Helmut Lütkepohl, No. 42 - March 1990.

A Geometrical Expository Note on Hausman's Specification Test. Howard E. Doran, No. 43 - March 1990.

Using The Kalman Filter to Estimate Sub-Populations. Howard E. Doran, No. 44 - March 1990.

Constraining Kalman Filter and Smoothing Estimates to Satisfy Time-Varying Restrictions. Howard Doran, No. 45 - May, 1990.

Multiple Minima in the Estimation of Models with Autoregressive Disturbances. Howard Doran and Jan Kmenta, No. 46 - May, 1990.

A Method for the Computation of Standard Errors for Geary Khamis-Parities and International Parities and International Prices. D.S. Prasada Rao and E.A. Selvanathan, No. 47 - September, 1990.

Prediction of the Probability of Successful First Year University Studies in Terms of High School Background: With Application to the Faculty of Economic Studies at the University of New England. D.M. Dancer and H.E. Doran, No. 48 - September, 1990.

A Generalized Theil-Tornqvist Index for Multilateral Comparisons. D.S. Prasada Rao and E.A. Selvanathan, No. 49 - November, 1990.

Frontier Production Functions and Technical Efficiency: A Survey of Empirical Applications in Agricultural Economics. George E. Battese, No. 50 - May 1991.

Consistent OLS Covariance Estimator and Misspecification Test for Models with Stationary Errors of Unspecified Form. Howard E. Doran, No. 51 - May 1991.

Testing Non-Nested Models. Howard E. Doran, No. 52 - May 1991.

Estimation of Australian Wool and Lamb Production Technologies: An Error Components Approach. C.J. O'Donnell and A.D. Woodland, No. 53 - October 1991.

Competitiveness Indices and the Trade Performance of the Australian Manufacturing Sector. C. Hargreaves, J. Harrington and A.M. Siriwardarna, No. 54 - October, 1991.

Modelling Money Demand in Australian Economy-Wide Models: Some Preliminary Analyses. Colin Hargreaves, No. 55 - October 1991.

Frontier Production Functions, Technical Efficiency and Panel Data: With Application to Paddy Farmers in India. G.E. Battese and T.J. Coelli, No. 56 - November 1991.