
T. J. Coelli

No. 57 - October 1991

ISSN 0.157-0188

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).

\[ Y_{it} = f(X_{it}; \beta) \cdot e^{(V_{it} - U_{it})}, \quad i=1, \ldots, N, \quad t=1, \ldots, T, \]

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,

\( \beta \) is a \( k \) by \( 1 \) vector of unknown parameters,

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

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

and the panel of data need not be complete.

We utilise the parameterization of Battese and Corra (1977) who replace \( \sigma_v^2 \) and \( \sigma^2 \) 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, \( \gamma \), 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 \( \eta \) 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 \( \mu \) 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 $\mu=0$ 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 preferred 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 $\gamma$. If the null hypothesis, that $\gamma$ equals zero, is accepted, then the ordinary least-squares model would be preferred 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.

\[ \ln Y_{it} = \beta_0 + \beta_1 \ln X_{i1t} + \ldots + \beta_k \ln X_{ikt} + V_{it} - U_{it} \]

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) \( X_{i1t} \)
   :
   :
  k+3) \( X_{ikt} \)

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 regressor 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. The three steps are:

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

2) A two-phase grid search of the $\gamma$, $\mu$ and $\eta$ parameters is conducted with the $\beta$-parameters (excepting $\beta_0$) set to the OLS values and the $\beta_0$ and $\sigma_s^2$ 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, $\gamma$, $\mu$ and $\eta$, if no restrictions are imposed upon the model. The values of $\beta_0$ and $\sigma_s^2$ 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:

- $\gamma$: from 0.1 to 0.9 in steps of GRIDNO
- $\mu$: from $-2.\sigma_{ols}$ to $2.\sigma_{ols}$ in steps of $2.\sigma_{ols}.GRIDNO$
- $\eta$: from $-GWIDTH$ to $GWIDTH$ in steps of $2.GWIDTH.GRIDNO$

where $\sigma_{ols}$ is the standard error of the OLS regression conducted in step (1) and the values of GRIDNO and GWIDTH are set in the file FRONT2.000 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 $\eta$ 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 $\mu$ and/or $\eta$ 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 \( \eta \) 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 EGI.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 \( \sigma_{v^2} \) and \( \sigma^2 \) imply that:

\[
\begin{align*}
\sigma_{S^2} &= \sigma_{v^2} + \sigma^2 = 0.04 + 0.09 = 0.13 \\
\gamma &= \frac{\sigma^2}{\sigma_{S^2}} = \frac{0.09}{0.13} = 0.69.
\end{align*}
\]

The data file, EGI.DTA, is listed in Table 1. Note that the columns are: firm number, time period, \( Y_{tt} \), \( X_{1tt} \) and \( X_{2tt} \), from left to right. We prefer to use FRONTIER in batch mode rather than interactively. Hence, the instruction file, EGI.INS, was constructed by making a copy of the file, BLANK.INS, and inserting the relevant information using a text editor. The instruction file, EGI.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 \( \mu \) 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:

<table>
<thead>
<tr>
<th>Model</th>
<th>Restrictions</th>
<th>Log-likelihood value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mu=0, \eta=0$</td>
<td>-12.45</td>
</tr>
<tr>
<td>2</td>
<td>$\eta=0$</td>
<td>-12.03</td>
</tr>
<tr>
<td>3</td>
<td>$\mu=0$</td>
<td>-8.26</td>
</tr>
<tr>
<td>4</td>
<td>no restrictions</td>
<td>-8.03</td>
</tr>
<tr>
<td>OLS</td>
<td>$\mu=0, \eta=0, \gamma=0$</td>
<td>-29.18</td>
</tr>
</tbody>
</table>

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

The output file, EGI.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 (egl.dta)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>14.96840</th>
<th>9.416352</th>
<th>35.13364</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>16.19860</td>
<td>4.643334</td>
<td>77.29733</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>9.185659</td>
<td>5.094714</td>
<td>89.79895</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5.651365</td>
<td>8.716519</td>
<td>27.87824</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>8.155598</td>
<td>1.066191</td>
<td>92.17432</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6.652379</td>
<td>0.258073</td>
<td>97.90739</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>18.07955</td>
<td>6.33816</td>
<td>82.08298</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2.789212</td>
<td>2.350246</td>
<td>38.87622</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>11.67540</td>
<td>1.076388</td>
<td>81.76085</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>3.565924</td>
<td>3.431948</td>
<td>9.475523</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>13.99463</td>
<td>4.032543</td>
<td>55.09601</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>20.61785</td>
<td>7.975470</td>
<td>73.12965</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>6.025520</td>
<td>0.3438121</td>
<td>65.38025</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>12.70686</td>
<td>2.440149</td>
<td>63.83928</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>23.87556</td>
<td>7.890738</td>
<td>59.24076</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>6.965972</td>
<td>2.906033</td>
<td>72.57392</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>9.319360</td>
<td>2.668065</td>
<td>68.91608</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>9.426871</td>
<td>4.219816</td>
<td>57.42408</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>17.91515</td>
<td>2.661315</td>
<td>87.84301</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>11.26414</td>
<td>2.454655</td>
<td>30.78941</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>12.17137</td>
<td>2.827333</td>
<td>93.73409</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>2.140435</td>
<td>0.4387420</td>
<td>35.96098</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>27.90314</td>
<td>6.751591</td>
<td>80.27460</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>10.71752</td>
<td>4.424754</td>
<td>49.88585</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>5.042543</td>
<td>1.583060</td>
<td>22.07209</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>6.010864</td>
<td>0.9072119</td>
<td>38.72705</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>3.702737</td>
<td>6.148555</td>
<td>5.322397</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>2.353757</td>
<td>0.4793847</td>
<td>2.520496</td>
</tr>
<tr>
<td>28</td>
<td>2</td>
<td>5.597053</td>
<td>1.954877</td>
<td>41.54480</td>
</tr>
<tr>
<td>29</td>
<td>2</td>
<td>9.962295</td>
<td>8.169494</td>
<td>68.38918</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>14.37167</td>
<td>4.054660</td>
<td>77.55609</td>
</tr>
<tr>
<td>31</td>
<td>2</td>
<td>21.57833</td>
<td>5.028795</td>
<td>77.81173</td>
</tr>
<tr>
<td>32</td>
<td>2</td>
<td>25.25393</td>
<td>6.937140</td>
<td>98.90436</td>
</tr>
<tr>
<td>33</td>
<td>2</td>
<td>13.87632</td>
<td>8.405021</td>
<td>42.73973</td>
</tr>
<tr>
<td>34</td>
<td>2</td>
<td>4.763233</td>
<td>2.502908</td>
<td>59.74079</td>
</tr>
<tr>
<td>35</td>
<td>2</td>
<td>9.226517</td>
<td>6.589737</td>
<td>18.08550</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
<td>9.517958</td>
<td>7.148683</td>
<td>26.65066</td>
</tr>
<tr>
<td>37</td>
<td>2</td>
<td>12.59230</td>
<td>8.039665</td>
<td>50.48774</td>
</tr>
<tr>
<td>38</td>
<td>2</td>
<td>17.54803</td>
<td>4.896381</td>
<td>88.18217</td>
</tr>
<tr>
<td>39</td>
<td>2</td>
<td>14.56668</td>
<td>6.721718</td>
<td>30.45071</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>19.92185</td>
<td>4.195113</td>
<td>95.83395</td>
</tr>
<tr>
<td>41</td>
<td>2</td>
<td>10.04695</td>
<td>4.550846</td>
<td>36.70369</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>38.97824</td>
<td>7.223219</td>
<td>89.31245</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>9.570093</td>
<td>4.870962</td>
<td>50.01818</td>
</tr>
<tr>
<td>44</td>
<td>2</td>
<td>9.430373</td>
<td>9.311944</td>
<td>40.99634</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>12.93405</td>
<td>2.894831</td>
<td>63.05118</td>
</tr>
<tr>
<td>46</td>
<td>2</td>
<td>17.50535</td>
<td>8.085479</td>
<td>60.99176</td>
</tr>
<tr>
<td>47</td>
<td>2</td>
<td>19.63722</td>
<td>8.656345</td>
<td>94.15947</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
<td>6.765616</td>
<td>3.426778</td>
<td>39.31178</td>
</tr>
<tr>
<td>49</td>
<td>2</td>
<td>15.19294</td>
<td>1.918254</td>
<td>78.62762</td>
</tr>
</tbody>
</table>
### Table 1 - Data File (eg1.dta) continued

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

### Table 2 - Instruction File (eg1.ins)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>eg1.dta</td>
<td>DATA FILE NAME</td>
<td></td>
</tr>
<tr>
<td>eg1.out</td>
<td>OUTPUT FILE NAME</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>NUMBER OF FIRMS</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NUMBER OF TIME PERIODS</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>TOTAL NUMBER OF OBSERVATIONS</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NUMBER OF REGRESSOR VARIABLES</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>MU (Y/N)</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>ETA (Y/N)</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>STARTING VALUES (Y/N)</td>
<td></td>
</tr>
<tr>
<td>IF YES THEN</td>
<td>BO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B1 TO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SIGMA SQUARED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GAMMA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ETA</td>
<td></td>
</tr>
</tbody>
</table>
Table 3 - Output File (egl.out)

OUTPUT FROM THE PROGRAM FRONTIER (version 2.0)

THE OLS ESTIMATES ARE:

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>STANDARD-ERROR</th>
<th>T-RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERCEPT</td>
<td>0.10299895E+00</td>
<td>0.23648802E+00</td>
</tr>
<tr>
<td>X 1</td>
<td>0.37353653E+00</td>
<td>0.55941207E-01</td>
</tr>
<tr>
<td>X 2</td>
<td>0.51800965E+00</td>
<td>0.56523434E-01</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>STANDARD-ERROR</th>
<th>T-RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERCEPT</td>
<td>0.32758622E+00</td>
<td>0.16321176E+00</td>
</tr>
<tr>
<td>X 1</td>
<td>0.34074197E+00</td>
<td>0.38708264E-01</td>
</tr>
<tr>
<td>X 2</td>
<td>0.53242275E+00</td>
<td>0.36430275E-01</td>
</tr>
<tr>
<td>SIGMA-SQUARED</td>
<td>0.19081528E+00</td>
<td>0.71495156E-01</td>
</tr>
<tr>
<td>GAMMA</td>
<td>0.77760260E+00</td>
<td>0.72916930E-01</td>
</tr>
<tr>
<td>MU IS RESTRICTED TO BE ZERO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ETA</td>
<td>0.11394257E+00</td>
<td>0.44530695E-01</td>
</tr>
</tbody>
</table>

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 (egl.out) continued

COVARIANCE MATRIX:

<table>
<thead>
<tr>
<th></th>
<th>0.26638080E-01</th>
<th>-0.17313582E-02</th>
<th>-0.47332317E-02</th>
<th>0.41787345E-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.35773371E-02</td>
<td>-0.10175208E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.17313582E-02</td>
<td>0.14983297E-02</td>
<td>-0.10848577E-04</td>
<td>0.34952280E-03</td>
<td></td>
</tr>
<tr>
<td>0.51860871E-03</td>
<td>-0.61110319E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.47332317E-02</td>
<td>-0.10848577E-04</td>
<td>0.13271650E-02</td>
<td>0.87685684E-04</td>
<td></td>
</tr>
<tr>
<td>0.23217409E-03</td>
<td>0.14696762E-04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.41787345E-02</td>
<td>0.34952280E-03</td>
<td>0.87685684E-04</td>
<td>0.51115574E-02</td>
<td></td>
</tr>
<tr>
<td>0.45745094E-02</td>
<td>-0.10883522E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.35773371E-02</td>
<td>0.51860871E-03</td>
<td>0.23217409E-03</td>
<td>0.45745094E-02</td>
<td></td>
</tr>
<tr>
<td>0.53168786E-02</td>
<td>-0.13494765E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.10175208E-02</td>
<td>-0.61110319E-03</td>
<td>0.14696762E-04</td>
<td>-0.10889522E-02</td>
<td></td>
</tr>
<tr>
<td>-0.13494765E-02</td>
<td>0.19813266E-02</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 1:

<table>
<thead>
<tr>
<th>FIRM-NO.</th>
<th>TECH.-EFF.-EST.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.64171075E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.93719961E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.35867279E+00</td>
</tr>
<tr>
<td>4</td>
<td>NO OBSERVATION IN THIS PERIOD</td>
</tr>
<tr>
<td>5</td>
<td>0.40155731E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.74762573E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.77161616E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.57133789E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.25591661E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.77678745E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.49469205E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.71318020E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.62810524E+00</td>
</tr>
<tr>
<td>14</td>
<td>NO OBSERVATION IN THIS PERIOD</td>
</tr>
<tr>
<td>15</td>
<td>0.75193711E+00</td>
</tr>
</tbody>
</table>

MEAN TECHNICAL EFF. = 0.65362018E+00
Table 3 - Output File (eg1.out) continued

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 2:

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

MEAN TECHNICAL EFF. = 0.68074659E+00

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 3:

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

MEAN TECHNICAL EFF. = 0.70656806E+00
Table 3 - Output File (eg1.out) continued

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 4:

<table>
<thead>
<tr>
<th>FIRM-NO.</th>
<th>TECH.-EFF.-EST.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.72873798E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.95468323E+00</td>
</tr>
<tr>
<td>3</td>
<td>NO OBSERVATION IN THIS PERIOD</td>
</tr>
<tr>
<td>4</td>
<td>0.56778587E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.52230613E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.81231923E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.83078618E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.67101232E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.37924706E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.83452688E+00</td>
</tr>
<tr>
<td>11</td>
<td>0.60533093E+00</td>
</tr>
<tr>
<td>12</td>
<td>0.78552479E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.71772647E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.68551600E+00</td>
</tr>
<tr>
<td>15</td>
<td>0.81564927E+00</td>
</tr>
</tbody>
</table>

MEAN TECHNICAL EFF. = 0.73100413E+00

TECHNICAL EFFICIENCY ESTIMATES FOR YEAR 5:

<table>
<thead>
<tr>
<th>FIRM-NO.</th>
<th>TECH.-EFF.-EST.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.75377508E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.95939573E+00</td>
</tr>
<tr>
<td>3</td>
<td>NO OBSERVATION IN THIS PERIOD</td>
</tr>
<tr>
<td>4</td>
<td>0.60321806E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.55999999E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.83046877E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.84730235E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.70026197E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.42086124E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.85065441E+00</td>
</tr>
<tr>
<td>11</td>
<td>NO OBSERVATION IN THIS PERIOD</td>
</tr>
<tr>
<td>12</td>
<td>0.80597622E+00</td>
</tr>
<tr>
<td>13</td>
<td>0.74360346E+00</td>
</tr>
<tr>
<td>14</td>
<td>0.71366562E+00</td>
</tr>
<tr>
<td>15</td>
<td>0.83350701E+00</td>
</tr>
</tbody>
</table>

MEAN TECHNICAL EFF. = 0.75400412E+00
Table 3 - Output File (egl.out) continued

<table>
<thead>
<tr>
<th>T:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

13 13 15 14 13 68

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

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


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


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)

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IPRINT - 1=PRINT ITERATIONS, 0=DO NOT</td>
</tr>
<tr>
<td>1</td>
<td>INDIC - USED IN UNIDIMENSIONAL SEARCH - SEE BELOW</td>
</tr>
<tr>
<td>0.00001</td>
<td>TOL - CONVERGENCE TOLERANCE (PROPORTIONAL)</td>
</tr>
<tr>
<td>0.001</td>
<td>TOL2 - TOLERANCE USED IN UNI-DIMENSIONAL SEARCH</td>
</tr>
<tr>
<td>1.0D+15</td>
<td>BIGNUM - USED TO SET BOUNDS ON DEN &amp; DIST</td>
</tr>
<tr>
<td>0.0001</td>
<td>STEP1 - SIZE OF 1ST STEP IN SEARCH PROCEDURE</td>
</tr>
<tr>
<td>1</td>
<td>IGRID2 - 1=DOUBLE ACCURACY GRID SEARCH, 0=SINGLE</td>
</tr>
<tr>
<td>0.1</td>
<td>GRIDNO - STEPS TAKEN IN 1ST PHASE GRID SEARCH ON GAMMA</td>
</tr>
<tr>
<td>1.0</td>
<td>GWIDTH - WIDTH EACH SIDE OF 0 OF GRID SEARCH ON ETA</td>
</tr>
<tr>
<td>0</td>
<td>IBBHH - 1=BERNT HALL HALL &amp; HOUSMAN S.E. ESTIMATES USED</td>
</tr>
<tr>
<td>100</td>
<td>MAXIT - MAXIMUM NUMBER OF ITERATIONS PERMITTED</td>
</tr>
</tbody>
</table>

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 specify 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 $1.0 \times 10^{15}$ 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 $\gamma$. It will also indirectly influence the degree of precision in the grid search on $\mu$ and $\eta$. 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 $\eta$ 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 $\eta$. 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 calculate 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 CONVG 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 $\gamma$, $\mu$ and $\eta$ (if no restrictions imposed). The parameters $\beta_0$ and $\sigma_s^2$ 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 $\sigma_s^2$ in the grid search.

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

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

ZI:
Calculates the value of $Z_i^*$ as defined in Battese and Coelli (1991).
EPE:
Calculates the product of the vector, $\eta_i$, and its transpose, as defined in Battese and Coelli (1991). EPE stands for 'Eta Prime Eta'.

EPR:
Calculates the product of the vector, $\eta_i$, and the transpose of the residual vector of the i-th firm. EPR stands for 'Eta Prime Residual'.
This program uses the Davidon-Fletcher-Powell algorithm to estimate using (unbalanced) panel data a frontier production function which has a technical efficiency distribution with a non-zero mode $\mu$ and a parameter $\eta$ to allow time-varying technical efficiency (a model formulated by Battese and Coelli (1991)).

A large proportion of the SEARCH, CONVRG, MINI and ETA subroutines are taken from Himmelblau (1972, appendix b). The functions DIS and DENS are based upon subroutines in Corra (1976, appendix B). The remainder of the program is the work of Tim Coelli.

Any person is welcome to copy and use this program free of charge. The author takes no responsibility for any inconvenience caused by undetected errors. If an error is detected the author would appreciate being informed.

A 1991 paper in the working paper series in the Department of Econometrics, University of New England, Armidale, NSW 2351, Australia describes the use of this program.

This version of FRONTIER is slightly different to earlier copies distributed for mainframes. It has been altered to run on an IBM PC/AT (and compatibles) with 640K using Lahey Fortran 77.

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(KI), 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,*) 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(KI=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(KI), XX(K1,K2,K3), YY(K1,K2), NN, NT, NP, NOMU, NOETA, NB

COMMON/SEVEN/GB(K4), S(V(K4), OBSE(K4)

DIMENSION GY(K4)

DO 98 I=1,K4
   GX(I)=0.0
   GY(I)=0.0
98 CONTINUE

IF (IGRID.EQ.I) THEN
   WRITE(6,'(A)') 'CALCULATING OLS ESTIMATES...'
   WRITE(6,'(A)') '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,'(A)') '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)
   CALL SEARCH
   ITER=ITER+1
   IF (ITER.GE.MAXIT) THEN
      WRITE(6,'(A)') '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

31
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=I,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 = ',15,' FUNC EVALS = ',I5,' LLF = ',E16.8)
302 FORMAT(4X,SEIS.8)
RETURN
END

SUBROUTINE CONVRG(IPASS)
Tests the convergence criterion.
The program is halted when the proportional change in the log-
likelihood and in each of the parameters is no greater than
a specified tolerance.
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER(KI=IOO,K2=20,K3=I6,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
40   IF(DABS(X(I)-Y(I)).GT.XTOL) GOTO 60
   CONTINUE
   IPASS=1
RETURN
SUBROUTINE ETA
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),DELDX(K4),DELG(K4),GX(K4)
COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,IGRID,MIXIT
COMMON/FIVE/TOL,TOL2,BIGNL,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+DELDX(I)*DELG(I)
20 DGHDG=DGHDG+DGH(I)*DELG(I)
DO 30 I=1,N
30 H(I,J)=H(I,J)+DELDX(I)*DELDX(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)=DELDX(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,NFUNCTION,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
COMMON/FIVE/TOL,TOL2,BIGNUM,STEP1,IGRID2,GGRIDNO,GRIDTH,IBHHHH
IEXIT=0
NTOL=0
FTOL=FTOL2
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)**2
102 SNORM=SNORM+S(I)**2
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 FA=F
   DB=D
   IF(K) 7,8,9
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
29 IF(F-FB) 29,13,13
30 IF(M) 31,31,13
31 M=M+1
32 DO 99 I=1,N
33 IF(Y(I).NE.X(I)) GOTO 325
99 CONTINUE
32 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
101 CONTINUE
WRITE (6,5000)
5000 FORMAT(' SEARCH FAILED ON GRADIENT STEP, TERMINATION' )
RETURN
33 IF(NTOL.EQ.5) GOTO 34
34 IF(IPRINT. EQ. I) 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(I.O-DIS(-ZI(I,B)))
DO 133 L=I,NP
IF (XX(I,L,1).NE.0.0) THEN
E=YY(I,L)
DO 134 J=I,NB
E=E-B(J)'XX(I,L,J)
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)
Calculates the fist-order partial derivatives of the negative
of the log-likelihood function and is also used to calculate
the i-th first-order partial derivative of the joint density
function of Y (used in the calculation of the covariance matrix
suggested by Berndt, Hall, Hall and Housman when the IBHHH flag is
set to 1 in the start-up file FRONT2.000)
IMPLIED 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 = I, NP
      E = YY(I, L)
      DO 135 K = I, 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) * (1.0 - B(N2)))
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
DO 136 I = KK, MM
A = A + 0.5 * (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
\[ A = A - 0.5 \times SS/(1.0 - B(N2))^2 \]  
\[ \text{GX}(N1) = A \]  

\[ A = A - 0.5 \times (F_\text{TOT} - F)/(1.0 - B(N2)) \]  
\[ \text{DO 140 } I = KK, MM \]  
\[ A = A + 0.5 \times (\text{EPE}(I,B) - I.0)/(I.0 + (\text{EPE}(I,B) - I.0)B(N2)) \]  
\[ \text{D} = B(N2) \times (1 - B(N2))^2 \times B(N1) \]  
\[ \text{DZI} = -(B(N3) + \text{EPR}(I,B)) \times D \]  
\[ \text{C} = 0.5 \times (B(N3) - 1.0) \times B(N2) \times \text{EPR}(I,B) \]  
\[ \text{DZI} = \text{DZI} - C \times (I.0 - 2.0 \times B(N2)) \times B(N2) \times (2.0 - 3.0 \times B(N2)) \]  
\[ \text{DZI} = \text{DZI} / (D^2 \times 1.5 \times B(N1))^0.5 \]  
\[ A = A - (\text{DEN}(-ZI(I,B))/(1.0 - \text{DIS}(-ZI(I,B)) + ZI(I,B)) \times \text{DZI} \]  

140 CONTINUE  
\[ \text{GX}(N2) = A \]  

\[ \text{IF (NOMU.EQ.1) THEN} \]  
\[ A = F/(B(N1)B(N2))^0.5 \times \text{DEN}(-Z)/(1.0 - \text{DIS}(-Z)) + Z \]  
\[ \text{DO 142 } I = KK, MM \]  
\[ \text{D} = \text{DEN}(-ZI(I,B))/(1.0 - \text{DIS}(-ZI(I,B)) + ZI(I,B)) \times (1.0 - B(N2)) \]  
\[ A = A - D/(B(N2) \times (1.0 - B(N2))^2 \times B(N1) \times (1.0 + (\text{EPE}(I,B) - I.0)B(N2)))^0.5 \]  

142 CONTINUE  
\[ \text{GX}(N3) = A \]  

END IF  

\[ \text{IF (NOETA.EQ.1) THEN} \]  
\[ A = 0.0 \]  
\[ \text{DO 151 } I = KK, MM \]  
\[ \text{DE} = 0.0 \]  
\[ \text{DO 152 } L = 1, NP \]  
\[ \text{IF (XX(I,L,1).EQ.1) THEN} \]  
\[ T = \text{DFLOAT}(L) \]  
\[ \text{DE} = \text{DE} - 2.0 \times (T - 1.0) \times \text{DEXP}(-2.0 \times B(N4) \times (T - \text{FNP})) \]  
\[ \text{E} = \text{YY}(I,L) \]  
\[ \text{DO 153 } J = 1, NB \]  
\[ \text{E} = \text{E} - \text{XX}(I,L,J) \times B(J) \]  

153 CONTINUE  
\[ \text{D} = \text{D} + (T - 1.0) \times \text{DEXP}(-B(N4) \times (T - \text{FNP})) \times \text{E} \]  

END IF  

152 CONTINUE  
\[ \text{DD} = (B(N2) \times (1.0 - B(N2)) \times B(N1) \times (1.0 + (\text{EPE}(I,B) - 1.0) \times B(N2))) \]  
\[ \text{D} = D \times B(N2) \]  
\[ \text{C} = B(N3) \times (1.0 - B(N2)) - B(N2) \times \text{EPR}(I,B) \]  
\[ \text{C} = C \times 0.5 \times B(N2)^2 \times (1.0 - B(N2)) \times B(N1) \times \text{DE} \]  
\[ \text{DZI} = (D - C) / (D^2 \times 1.5) \]  
\[ A = A - (\text{DEN}(-ZI(I,B))/(1.0 - \text{DIS}(-ZI(I,B)) + ZI(I,B)) \times \text{DZI} \]  
\[ A = A + \text{B}(N2) \times 2.0 \times \text{DE} / (1.0 + (\text{EPE}(I,B) - 1.0) \times B(N2)) \]  

151 CONTINUE  
\[ \text{GX}(N4) = A \]  

END IF  

NDRV = NDRV + 1
DOUBLE PRECISION FUNCTION DEN(A)
Evaluates the N(0,1) density function.

DOUBLE PRECISION FUNCTION DIS(X)
Evaluates the N(0,1) distribution function.
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)
Calculates the value of zi as defined in Battese and Coelli (1991)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER(KI=100,K2=20,K3=16,K4=K3+4)
COMMON/THREE/N,NFUNCT,NDRV,ITER,INDIC,IPRINT,GRID,MXIT
   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)
Calculates eta prime eta.
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER(KI=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.I) THEN
      N4=NB+4
   ELSE
      N4=NB+3
   END IF
   FNP=DFLOAT(NP)
   DO I01=I,NP
      IF(XX(I,L,1).NE.0.0) TMP=TMP+DEXP(-2.0-B(N4)-(DFLOAT(L)-FNP))
   I01
   EPE=TMP
   END

DOUBLE PRECISION FUNCTION EPR(I,B)
Calculates eta prime residual.
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER(KI=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)
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
      ENDIF
      WRITE(6,*)'HOW MANY TIME PERIODS IN THE DATA ? '
   ENDIF
WRITE(6,*)
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 (NT.NP.GT.0) 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,*) ' DO YOU WISH TO SUPPLY STARTING VALUES ? (Y or N) '
READ(5,*) CHST
IF ( (CHST.EQ. 'Y').OR. (CHST.EQ. 'y') ) THEN
GRID=0
62 FORMAT(' ENTER STARTING VALUE FOR B', I2, ' : ' )
DO 138 I=1,NB+1
WRITE(6,62) I-1
READ(5,*) Y(I)
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
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)
      CONTINUE
      IF ((CHMU.EQ.'Y').OR.(CHMU.EQ.'y')) THEN
        READ(50,*) Y(NB+4)
      ELSE
        READ(50,*) Y(NB+4)
      END IF
      ELSE
        WRITE(6,*) ' INCORRECT OPTION - MUST BE T OR F - BYE!' STOP
      END IF
      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.KI) THEN
        WRITE(6,70) KI
        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=I
      IF ((CHETA.EQ.'Y').OR.(CHETA.EQ.'y')) NOETA=I
      NB=NB+I
      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.
C
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,NFUNCTION,NDRV,ITER,INDIC,IPRINT,IGRID,MAXIT
C
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/OF(K4),OB(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 (GRID.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(' INTERCEPT 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(N1)
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)
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,1)**0.5,Y(I)/H(I,1)**0.5
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 )
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
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*$ABS(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 = ',I11)
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,',''
END IF
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)
CONTINUE
IF (IBHHH.EQ.1) THEN
WRITE(70,*)
WRITE(70,'(NOTE: THIS IS THE BHNN 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).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))^2'/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(-ETA*(1.0-DIS(-FI/SI)))*Y(N3)'(1.0-Y(N2))**0.5*Y(N1)*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 I=1,NN
MT(L)=L
449 CONTINUE
WRITE(70,442) (MT(L),L=1,NP)
C FIXFORMAT
442 FORMAT(10214)
450 I=1,NN
WRITE(70,443) I,(INT(XX(I,L,1)),L=1,NP),M(I)
450 CONTINUE
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,NFUNCTION,DVALUE,ITER,INDIC,IPRINT,IGRID,MAXIT
COMON/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)
FN=DFLOAT(NT)
CALL OLS(OB,OBSE)
DO 131 I=I,N
Y(I)=OB(I)
131 CONTINUE
B0=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 135 I=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)=BO+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/(1.0 + (R*((PI-2.0)/PI)-1.0)*Y(N2))
Z = Y(N3)/Y(N2)*Y(N1)**0.5
Z = (Y(N2)*Y(N1)**0.5*DEN(-Z)/(1.0-DIS(-Z))
Y(1) = BO + Y(N3) + Z
IF(Y(1).LT.(BO+DEV)) THEN
CALL FUN(Y, FY)
IF(FY.LT.FX) THEN
DO 141 I = 1, N
X(I) = Y(I)
141 CONTINUE
END IF
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 AMAX = DABS(XX(J,J))
5 ICOL = J
AMAX = DABS(XX(J,J))
4 CONTINUE
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
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,NI,NOMU,NOE,NA
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,NI
            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,NI
            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=(I.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
   DO 122 I=1,NB
      DO 122 J=1,NB
         XPX(I,J)=XPX(I,J)*10.0**(MX(I)+MX(J))
   122 CONTINUE
   WRITE(6,*),IS
   RETURN
END
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


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


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

Estimation of Frontier Production Functions and the Efficiencies of Indian
Farms Using Panel Data from ICRISAT's Village Level Studies.

Estimation of Frontier Production Functions: A Guide to the Computer Program,

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,

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 -

A Mixed Integer Linear Programming Evaluation of Salinity and Waterlogging
Control Options in the Murray-Darling Basin of Australia.

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

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 Seary Khamis-Parities and
International Parities and International Prices. D.S. Prasada Rao and
E.A. Selvanathan, No. 47 - September, 1990.


