Appendices

Appendix 1: Derivations for the Time-varying Inefficiency Model

Consider the frontier production function[1]

$$Y_{it} = x_{it}\beta + E_{it} \tag{A1.1}$$

where

$$E_{it} = V_{it} - \eta_{it}U_i \tag{A1.2}$$

and

$$\eta_{it} = e^{-\eta(t-T)}, \quad t \in \vartheta(i); \; i = 1,2,\ldots,N. \tag{A1.3}$$

It is assumed that the V_{it}s are independent and identically distributed $N(0,\sigma_V^2)$ random variables, independent of the U_is, which are assumed to be non-negative truncations of the $N(\mu,\sigma^2)$ distribution.

The density function for U_i is

$$f_{U_i}(u_i) = \frac{\exp\left[-\frac{1}{2}(u_i - \mu)^2 / \sigma^2\right]}{(2\pi)^{1/2}\sigma\left[1 - \Phi(-\mu/\sigma)\right]}, \quad u_i \geq 0, \tag{A1.4}$$

where $\Phi(\cdot)$ represents the distribution function for the standard normal random variable.

It can be shown that the mean and variance of U_i are[2]

---

[1] In the frontier model (3.2), the notation, Y_{it}, represented the actual production at the time of the t-th observation for the i-th firm. However, given that (3.2) involves a Cobb-Douglas or transcendental logarithmic model, then Y_{it} and x_{it} in this Appendix would represent logarithms of output and input values, respectively.

[2] We prefer not to use the notation, σ_U^2, for the variance of the normal distribution which is truncated (at zero) to obtain the distribution of the non-negative technical inefficiency effects, because this variance is <u>not</u> the variance of U_i . For the case of the half-normal distribution the variance of U_i is $\sigma^2(\pi-2)/\pi$. This fact needs to be kept in mind in the interpretation of empirical results for the stochastic frontier model.

$$E(U_i) = \mu + \sigma\left\{\phi(-\mu/\sigma)/\left[1-\Phi(-\mu/\sigma)\right]\right\} \tag{A1.5}$$

and

$$\mathrm{Var}(U_i) = \sigma^2\left\{1-\frac{\phi(-\mu/\sigma)}{1-\Phi(-\mu/\sigma)}\left[\frac{\mu}{\sigma}+\frac{\phi(-\mu/\sigma)}{1-\Phi(-\mu/\sigma)}\right]\right\}, \tag{A1.6}$$

where $\phi(\cdot)$ represents the density function for the standard normal distribution.

From the joint density function for U_i and V_i, where V_i represents the $(T_i \times 1)$ vector of the V_{it}s associated with the T_i observations for the i-th firm, it follows readily that the joint density function for U_i and E_i, where E_i is the $(T_i \times 1)$ vector of the values of $E_{it} \equiv V_{it} - \eta_{it}U_i$, is

$$f_{U_i,E_i}(u_i,e_i) = \frac{\exp-\frac{1}{2}\left\{\left[(u_i-\mu)^2/\sigma^2\right]+\left[(e_i+\eta_i u_i)'(e_i+\eta_i u_i)/\sigma_V^2\right]\right\}}{(2\pi)^{(T_i+1)/2}\,\sigma\,\sigma_V{}^{T_i}\left[1-\Phi(-\mu/\sigma)\right]} \tag{A1.7}$$

where e_i is a possible value for the random vector, E_i.

The density function for E_i, obtained by integrating $f_{U_i,E_i}(u_i,e_i)$ with respect to the range for U_i, namely $u_i \geq 0$, is

$$f_{E_i}(e_i) = \frac{\left[1-\Phi\left(-\mu_i^*/\sigma_i^*\right)\right]\exp-\frac{1}{2}\left\{\left(e_i'e_i/\sigma_V^2\right)+(\mu/\sigma)^2-(\mu_i^*/\sigma_i^*)^2\right\}}{(2\pi)^{T_i/2}\,\sigma_V{}^{(T_i-1)}\left[\sigma_V^2+\eta_i'\eta_i\sigma^2\right]^{1/2}\left[1-\Phi(-\mu/\sigma)\right]} \tag{A1.8}$$

where $$\mu_i^* \equiv \frac{\mu\sigma_V^2-\eta_i'e_i\sigma^2}{\sigma_V^2+\eta_i'\eta_i\sigma^2} \tag{A1.9}$$

and $$\sigma_i^{*2} \equiv \frac{\sigma_V^2\sigma^2}{\sigma_V^2+\eta_i'\eta_i\sigma^2}. \tag{A1.10}$$

From the above results, it follows that the conditional density function of U_i, given that the random vector, E_i, has value, e_i, is

$$f_{U_i|E_i=e_i}(u_i) = \frac{\exp-\frac{1}{2}\left[(\mu_i-\mu_i^*)/\sigma_i^*\right]^2}{(2\pi)^{1/2}\,\sigma_i^*\left[1-\Phi\left(-\mu_i^*/\sigma_i^*\right)\right]}, \quad u_i \geq 0. \tag{A1.11}$$

This is the density function of the positive truncation of the $N(\mu_i^* / \sigma_i^{*2})$ distribution.

Since the conditional expectation of $\exp(-\eta_{it}U_i)$, given $E_i = e_i$, is defined by

$$E\left\{\exp\left(-\eta_{it}U_i|E_i = e_i\right)\right\} = \int_0^\infty \exp(-\eta_{it}u_i) f_{U_i|E_i=e_i}(u_i)du_i ,$$

the result of equation (3.3) is obtained by straightforward integral calculus.

If the frontier production function (A1.1)-(A1.3) is appropriate for production, expressed in the original units of output, then the prediction of the technical efficiency of the i-th firm at the time of the t-th observation, $TE_{it} = 1-(\eta_{it}U_i/x_{it}\beta)$, requires the conditional expectation of U_i, given $E_i = e_i$. This can be shown to be

$$E\left(U_i|E_i = e_i\right) = \mu_i^* + \sigma_i^*\left\{\phi\left(-\mu_i^* / \sigma_i^*\right) / \left[1 - \Phi\left(-\mu_i^* / \sigma_i^*\right)\right]\right\} \qquad \text{(A1.12)}$$

where μ_i^* and σ_i^{*2} are defined by equations (A1.9) and (A1.10), respectively.

The density function for Y_i, the $(T_i \times 1)$ random vector of Y_{it}s for the i-th firm, is obtained from (A1.8) by substituting $(y_i - x_i\beta)$ for e_i, where x_i is the $(T_i \times k)$ matrix of x_{it}s for the i-th firm, where k is the dimension of the vector, β. The logarithm of the likelihood function for the sample observations, $y \equiv (y'_1, y'_2, \ldots, y'_N)'$, is thus

$$\begin{aligned} L^*(\theta^*; y) = & -\tfrac{1}{2}\left(\sum_{i=1}^{N} T_i\right)\ell n(2\pi) - \tfrac{1}{2}\sum_{i=1}^{N}(T_i - 1)\ell n(\sigma_V^2) - \tfrac{1}{2}\sum_{i=1}^{N}\ell n(\sigma_V^2 + \eta_i'\eta_i\sigma^2) \\ & -N\,\ell n\left[1 - \Phi(-\mu / \sigma)\right] + \sum_{i=1}^{N}\ell n\left[1 - \Phi(-\mu_i^* / \sigma_i^*)\right] \\ & -\tfrac{1}{2}\sum_{i=1}^{N}\left[(y_i - x_i\beta)'(y_i - x_i\beta) / \sigma_V^2\right] - \tfrac{1}{2}N(\mu / \sigma)^2 + \tfrac{1}{2}\sum_{i=1}^{N}(\mu_i^* / \sigma_i^*)^2 \end{aligned}$$

(A1.13)

where $\theta^* \equiv (\beta', \sigma^2_V, \sigma^2, \mu, \eta)'$.

Using the reparameterisation of the model, suggested by Battese and Corra (1977), where $\sigma^2_V + \sigma^2 = \sigma^2_S$ and $\gamma = \sigma^2/\sigma^2_S$, the logarithm of the likelihood function is expressed by

$$L^*(\theta;y) = -\tfrac{1}{2}\left(\sum_{i=1}^{N} T_i\right)\left\{\ell n(2\pi) + \ell n(\sigma_S^2)\right\} - \tfrac{1}{2}\sum_{i=1}^{N}(T_i - 1)\ell n(1-\gamma)$$

$$-\tfrac{1}{2}\sum_{i=1}^{N}\ell n\left[1 + (\eta_i'\eta_i - 1)\gamma\right] - N\ell n\left[1 - \Phi(-z)\right] - \tfrac{1}{2}Nz^2$$

$$+\sum_{i=1}^{N}\ell n\left[1 - \Phi(-z_i^*)\right] + \tfrac{1}{2}\sum_{i=1}^{N} z_i^{*2} - \tfrac{1}{2}\sum_{i=1}^{N}(y_i - x_i\beta)'(y_i - x_i\beta)/(1-\gamma)\sigma_S^2,$$

(A1.14)

where $\theta \equiv (\beta', \sigma^2{}_S, \gamma, \mu, \eta)'$, $z \equiv \mu/(\gamma\sigma^2{}_S)^{1/2}$ and

$$z_i^* = \frac{\mu(1-\gamma) - \gamma\eta_i'(y_i - x_i\beta)}{\left\{\gamma(1-\gamma)\sigma_S^2\left[1 + (\eta_i'\eta_i - 1)\gamma\right]\right\}^{1/2}}.$$

The partial derivations of the loglikelihood function (A1.14) with respect to the parameters, β, $\sigma^2{}_S$, γ, μ and η, are given by

$$\frac{\partial L^*}{\partial \beta} = \sum_{i=1}^{N} x_i'(y_i - x_i\beta)\left[(1-\gamma)\sigma_S^2\right]^{-1}$$

$$+\sum_{i=1}^{N}\left[\frac{\phi(-z_i^*)}{1-\Phi(-z_i^*)} + z_i^*\right]\gamma\, x_i'\eta_i\left\{\gamma(1-\gamma)\sigma_S^2\left[1 + (\eta_i'\eta_i - 1)\gamma\right]\right\}^{-1/2}$$

$$\frac{\partial L^*}{\partial \sigma_S^2} = -\frac{1}{2\sigma_S^2}\left\{\sum_{i=1}^{N} T_i - N\left[\frac{\phi(-z)}{1-\Phi(-z)} + z\right]z + \sum_{i=1}^{N}\left[\frac{\phi(-z_i^*)}{1-\Phi(-z_i^*)} + z_i^*\right]z_i^*\right.$$

$$\left.-\sum_{i=1}^{N}(y_i - x_i\beta)'(y_i - x_i\beta)\left[(1-\gamma)\sigma_S^2\right]^{-1}\right\}$$

$$\frac{\partial L^*}{\partial \gamma} = \frac{(1-\gamma)^{-1}}{2}\sum_{i=1}^{N}(T_i - 1) - \frac{1}{2}\sum_{i=1}^{N}(\eta_i'\eta_i - 1)\left[1 + (\eta_i'\eta_i - 1)\gamma\right]^{-1}$$

$$+\frac{N}{2}\left[\frac{\phi(-z)}{1-\Phi(-z)} + z\right]z\gamma^{-1} + \sum_{i=1}^{N}\left[\frac{\phi(-z_i^*)}{1-\Phi(-z_i^*)} + z_i^*\right]\frac{\partial z_i^*}{\partial \gamma}$$

$$-\frac{1}{2}\sum_{i=1}^{N}(y_i - x_i\beta)'(y_i - x_i\beta)\left[(1-\gamma)\sigma_S\right]^{-2}$$

$$\frac{\partial L^*}{\partial \mu} = -\frac{N}{\left(\gamma\sigma_S^2\right)^{1/2}}\left[\frac{\phi(-z)}{1-\Phi(-z)} + z\right] + \sum_{i=1}^{N}\left[\frac{\phi\left(-z_i^*\right)}{1-\Phi\left(-z_i^*\right)} + z_i^*\right]$$

$$\times \frac{(1-\gamma)}{\left\{\gamma(1-\gamma)\sigma_S^2\left[1+\left(\eta_i'\eta_i - 1\right)\gamma\right]\right\}^{1/2}}$$

$$\frac{\partial L^*}{\partial \eta} = \sum_{i=1}^{N}\left[\frac{\phi\left(-z_i^*\right)}{1-\Phi\left(-z_i^*\right)} + z_i^*\right]\frac{\partial z_i^*}{\partial \eta} - \frac{\gamma}{2}\sum_{i=1}^{N}\frac{\partial \eta_i'\eta_i}{\partial \eta}\left[1+\left(\eta_i'\eta_i - 1\right)\gamma\right]^{-1}$$

$$\frac{\partial z_i^*}{\partial \gamma} = -\frac{\left[\mu + \eta_i'(y_i - x_i\beta)\right]}{\sigma_S\left\{\gamma(1-\gamma)\left[1+\left(\eta_i'\eta_i - 1\right)\gamma\right]\right\}^{1/2}}$$

$$-\frac{1}{2}\frac{\left[\mu(1-\gamma) - \gamma\eta_i'(y_i - x_i\beta)\right]\left[(1-2\gamma) + \left(\eta_i'\eta_i - 1\right)\gamma(2-3\gamma)\right]}{\sigma_S\left\{\gamma(1-\gamma)\left[1+\left(\eta_i'\eta_i - 1\right)\gamma\right]\right\}^{3/2}}$$

$$\frac{\partial z_i^*}{\partial \eta} = \frac{\gamma\sum_{t\in\vartheta(i)}(t-T)\,e^{-\eta(t-T)}(y_{it} - x_{it}\beta)}{\left\{\gamma(1-\gamma)\sigma_S^2\left[1+\left(\eta_i'\eta_i - 1\right)\gamma\right]\right\}^{1/2}} - \frac{\left[\mu(1-\gamma) - \gamma\eta_i'(y_i - x_i\beta)\right]\frac{1}{2}\gamma^2(1-\gamma)\sigma_S^2\frac{\partial \eta_i'\eta_i}{\partial \eta}}{\left\{\gamma(1-\gamma)\sigma_S^2\left[1+\left(\eta_i'\eta_i - 1\right)\gamma\right]\right\}^{3/2}}$$

and $\quad \dfrac{\partial \eta_i'\eta_i}{\partial \eta} = -2\sum_{t\in\vartheta(i)}(t-T)e^{-2\eta(t-T)} \qquad$ if $\eta \neq 0$.

Appendix 2: Derivations for the Stochastic Frontier and Inefficiency Model

For simplicity of presentation of results in this Appendix, we assume that the stochastic frontier and inefficiency model is expressed by

$$Y_{it} = x_{it}\beta + E_{it} \tag{A2.1}$$

and

$$E_{it} = V_{it} - U_{it}, \tag{A2.2}$$

where $i = 1,2,\ldots,N$, and $t = 1,2,\ldots,T$.

Thus, given the frontier production function of equation (4.1), Y_{it} in equation (A2.1) is, in fact, the logarithm of the production for the i-th firm in the t-th time period. Apart from introducing the notation, E_{it}, for the difference between V_{it} and U_{it}, all other variables and parameters in equations (A2.1) and (A2.2) are the same as in equations (4.1) and (4.2).

The density functions for V_{it} and U_{it} are

$$f_V(v) = \frac{\exp(-\frac{1}{2}v^2/\sigma_V^2)}{\sqrt{2\pi}\,\sigma_V}, \quad -\infty < v < \infty \tag{A2.3}$$

and

$$f_U(u) = \frac{\exp\left[-\frac{1}{2}(u - z\delta)^2/\sigma^2\right]}{\sqrt{2\pi}\,\sigma\,\Phi[z\delta/\sigma]}, \quad u \geq 0, \tag{A2.4}$$

where the subscripts, i and t, are omitted for convenience in the presentation.

The joint density function for E = V-U and U is

$$f_{E,U}(e,u) = \frac{\exp-\frac{1}{2}\left\{\left[(e+u)^2/\sigma_V^2\right]+\left[(u-z\delta)^2/\sigma^2\right]\right\}}{2\pi\,\sigma\,\sigma_V\,\Phi[z\delta/\sigma]}, \quad u \geq 0$$

$$= \frac{\exp-\frac{1}{2}\left\{\left[(u-\mu_*)^2/\sigma_*^2\right]+\left(e^2/\sigma_V^2\right)+(z\delta/\sigma)^2-(\mu_*/\sigma_*)^2\right\}}{2\pi\,\sigma\,\sigma_V\,\Phi[z\delta/\sigma]} \tag{A2.5a}$$

or, alternatively,

$$f_{E,U}(e,u) = \frac{\exp -\frac{1}{2}\left\{\left[(u-\mu_*)^2 / \sigma_*^2\right] + \left[(e+z\delta)^2 / \left(\sigma_V^2 + \sigma^2\right)\right]\right\}}{2\pi\, \sigma\, \sigma_V\, \Phi[z\delta / \sigma]} \tag{A2.5b}$$

where

$$\mu_* = \frac{\sigma_V^2 z\delta - \sigma^2 e}{\sigma_V^2 + \sigma^2} \tag{A2.6}$$

and

$$\sigma_*^2 = \sigma^2 \sigma_V^2 / \left(\sigma^2 + \sigma_V^2\right) . \tag{A2.7}$$

Thus the density function for E = V-U is

$$f_E(e) = \frac{\exp -\frac{1}{2}\left\{\left(e^2 / \sigma_V^2\right) + (z\delta / \sigma)^2 - (\mu_* / \sigma_*)^2\right\}}{\sqrt{2\pi}\, \sigma_V \sigma\, \Phi[z\delta / \sigma]} \int_0^\infty \frac{\exp -\frac{1}{2}\left[(u-\mu_*)^2 / \sigma_*^2\right]}{\sqrt{2\pi}} du$$

$$= \frac{\exp -\frac{1}{2}\left\{\left(e^2 / \sigma_V^2\right) + (z\delta / \sigma)^2 - (\mu_* / \sigma_*)^2\right\}}{\sqrt{2\pi}\left(\sigma^2 + \sigma_V^2\right)^{1/2}\left[\Phi(z\delta / \sigma) / \Phi(\mu_* / \sigma_*)\right]} , \tag{A2.8a}$$

or, alternatively,

$$f_E(e) = \frac{\exp\left\{-\frac{1}{2}(e+z\delta)^2 / \left(\sigma_V^2 + \sigma^2\right)\right\}}{\sqrt{2\pi}\left(\sigma_V^2 + \sigma^2\right)^{1/2}\left[\Phi(z\delta / \sigma) / \Phi(\mu_* / \sigma_*)\right]} . \tag{A2.8b}$$

The conditional density function for U given E = e is thus

$$f_{U|E=e}(u) = \frac{\exp -\frac{1}{2}\left[(u-\mu_*)^2 / \sigma_*^2\right]}{\sqrt{2\pi}\, \sigma_*\, \Phi(\mu_* / \sigma_*)}, \quad u \geq 0 . \tag{A2.9}$$

It can be shown that the conditional expectation of e^{-U}, given E = e, is

$$E\left(e^{-U} | E = e\right) = \left\{\exp\left[-\mu_* + \tfrac{1}{2}\sigma_*^2\right]\right\}\left\{\Phi\left[(\mu_* / \sigma_*) - \sigma^*\right] / \Phi(\mu_* / \sigma_*)\right\} . \tag{A2.10}$$

The density function for the production value, Y_{it}, in equation (A2.1), is most conveniently given using the expression in equation (A2.8b),

$$f_{Y_{it}}(y_{it}) = \frac{\exp -\frac{1}{2}\left\{\frac{(y_{it} - x_{it}\beta + z_{it}\delta)^2}{\sigma_V^2 + \sigma^2}\right\}}{\sqrt{2\pi}\left(\sigma_V^2 + \sigma^2\right)^{1/2}\left[\Phi(d_{it}) / \Phi(d_{it}^*)\right]} \tag{A2.11}$$

where $d_{it} = z_{it}\delta / \sigma$, $d_{it}^* = \mu_{it}^* / \sigma_*$ and $\mu_{it}^* = \left[\sigma_V^2\ z_{it}\delta - \sigma^2(y_{it} - x_{it}\beta)\right] / \left(\sigma_V^2 + \sigma^2\right)$.

Given that there are T_i observations obtained for the i-th firm, where $1 \leq T_i \leq T$, and $Y_i \equiv (Y_{i1}, Y_{i2}, \ldots, Y_{iT_i})'$ denotes the vector of the T_i production values in equation (A2.1), then the logarithm of the likelihood function for the sample observations, $y \equiv (y_1', y_2', \ldots, y'_T)'$, is

$$\begin{aligned} L^*(\theta^*; y) = & -\tfrac{1}{2}\left(\sum_{i=1}^{N} T_i\right)\left\{\ell n\ 2\pi + \ell n\left(\sigma^2 + \sigma_V^2\right)\right\} \\ & -\tfrac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\left[(y_{it} - x_{it}\beta + z_{it}\delta)^2 / \left(\sigma_V^2 + \sigma^2\right)\right] \\ & -\sum_{i=1}^{N}\sum_{t=1}^{T}\left[\ell n\ \Phi(d_{it}) - \ell n\ \Phi(d_{it}^*)\right] \end{aligned} \tag{A2.12}$$

where $\theta^* = \left(\beta', \delta', \sigma_V^2, \sigma^2\right)'$.

Using the re-parameterisation of the model involving the parameters, $\sigma_S^2 \equiv \sigma_V^2 + \sigma^2$ and $\gamma \equiv \sigma^2 / \sigma_S^2$, the logarithm of the likelihood function is expressed by

$$\begin{aligned} L^*(\theta; y) = & -\tfrac{1}{2}\left(\sum_{i=1}^{N} T_i\right)\left\{\ell n\ 2\pi + \ell n\ \sigma_S^2\right\} \\ & -\tfrac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\left\{(y_{it} - x_{it}\beta + z_{it}\delta)^2 / \sigma_S^2\right\} \\ & -\sum_{i=1}^{N}\sum_{t=1}^{T}\left\{\ell n\ \Phi(d_{it}) - \ell n\ \Phi(d_{it}^*)\right\} \end{aligned} \tag{A2.13}$$

where $d_{it} = z_{it}\delta / \left(\gamma\sigma_S^2\right)^{1/2}$ (A2.14)

$$d_{it}^* = \mu_{it}^* / \left[\gamma(1-\gamma)\sigma_S^2\right]^{1/2} \tag{A2.15}$$

$$\mu_{it}^* = (1-\gamma)z_{it}\delta - \gamma(y_{it} - x_{it}\beta) \tag{A2.16}$$

$$\sigma_* = \left[\gamma(1-\gamma)\sigma_S^2\right]^{1/2} \tag{A2.17}$$

and $\theta = \left(\beta', \delta', \sigma_S^2, \gamma\right)'$.

The partial derivatives of the logarithm of the likelihood function (A2.13) with respect to the parameters, β, δ, σ_S^2, and γ, are given by

$$\frac{\partial L^*}{\partial \beta} = \sum_{i=1}^{N}\sum_{t=1}^{T_i}\left\{\frac{(y_{it} - x_{it}\beta + z_{it}\delta)}{\sigma_S^2} + \frac{\phi(d_{it}^*)}{\Phi(d_{it}^*)}\frac{\gamma}{\sigma_*}\right\}x_{it}'$$

$$\frac{\partial L^*}{\partial \delta} = -\sum_{i=1}^{N}\sum_{t=1}^{T_i}\left\{\frac{(y_{it} - x_{it}\beta + z_{it}\delta)}{\sigma_S^2} + \left[\frac{\phi(d_{it})}{\Phi(d_{it})}\cdot\frac{1}{\left(\gamma\,\sigma_S^2\right)^{1/2}} - \frac{\phi(d_{it}^*)}{\Phi(d_{it}^*)}\cdot\frac{(1-\gamma)}{\sigma_*}\right]\right\}z_{it}'$$

$$\frac{\partial L^*}{\partial \sigma_S^2} = -\tfrac{1}{2}\left(\frac{1}{\sigma_S^2}\right)\left\{\left(\sum_{i=1}^{N}T_i\right) - \sum_{i=1}^{N}\sum_{t=1}^{T_i}\left[\frac{\phi(d_{it})}{\Phi(d_{it})}d_{it} - \frac{\phi(d_{it}^*)}{\Phi(d_{it}^*)}d_{it}^*\right]\right.$$
$$\left. - \sum_{i=1}^{N}\sum_{t=1}^{T_i}\frac{(y_{it} - x_{it}\beta + z_{it}\delta)}{\sigma_S^2}\right\}$$

$$\frac{\partial L^*}{\partial \gamma} = \sum_{i=1}^{N}\sum_{t=1}^{T_i}\left\{\frac{\phi(d_{it})}{\Phi(d_{it})}\frac{d_{it}}{2\gamma} + \frac{\phi(d_{it}^*)}{\Phi(d_{it}^*)}\left[\frac{y_{it} - x_{it}\beta + z_{it}\delta}{\sigma_*} + \frac{d_{it}^*(1-2\gamma)}{2\gamma(1-\gamma)\sigma_*^2}\right]\right\}.$$

Appendix 3: Electricity Data

Important details on the sources and definitions of data used in the study in Chapter 6 are discussed below.

Missing data:

Two plants began operation during the sample period. These were Loy Yang A and Bayswater, which were commissioned in 1983/84 and 1985/86, respectively.

Data on labour was not available for 1988/89 in the four Victorian plants because of a change over from local maintenance staff to a maintenance group which moved from plant to plant. This change made labour figures difficult to derive.

The NSW data were collected by Michael Plumb from annual reports for use in an undergraduate project at the University of Sydney, and was kindly supplied for this analysis. The data that were supplied did not include observations for 1981/82. Hence this year are not considered for the six NSW plants in this study.

Production data were not available for Bunbury in 1981/82. These were the only observations missing from the three WA plants.

Derived data:

Data on number of employees were not available from Victoria, but data on wages and salaries of employees were available. Hence the number of employees in Victorian plants was derived using an implicit wage series obtained from the WA data. The WA expenditure on employees was divided by the number of WA employees to obtain an implicit wage series. It was then assumed that similar wages would be paid in Victoria so that the expenditure on employees in Victorian plants could be divided by this wage series to obtain a measure of the number of employees in Victorian plants.

The average age of generating units is a weighted average of the ages of the units at a plant, where the weights are the name-plate capacities of the units.

The output measure is electricity sent out. Capacity factor is calculated using this figure, not the electricity generated.

The measure of total fuel usage in terrajoules was derived by converting each of the fuels into their energy equivalents. The energy contents assumed for the various fuels are listed in Table A3.1.

Table A3.1

Energy Contents of Fuels (in megajoules/kilogram) in 13 Power Plants in Australia

Plant	Coal	Oil	Briquettes
Loy Yang A	8.6		22.1
Hazelwood	8.8		22.1
Yallourn W	7.4	43.7	
Morwell	8.8	43.7	
Bayswater	23.4	45.9	
Eraring	23.9	45.9	
Liddell	22.2	46.4	
Munmorah	26.4	45.9	
Vales Point B	23.8	43.7	
Wallerang C	25.0	45.9	
Bunbury	20.0	43.0	
Muja	20.0	43.0	
Kwinana	19.9	43.0	

Table A3.2

Data From 13 Australian Coal-Fired Power Plants

Year	Plant	cap (mw)	output (kwh)	coal (000t)	gas (tj)	oil (000t)	briqts (000t)	lab (per)	tot-fuel (tj)	a/age (yrs)	a/size (mw)	c/fac (%)
83/84	loy	500	376	404.8	0	0.0	98.4	41	5656	0	500	8.58
84/85	loy	1000	4057	5007.4	0	0.0	87.8	458	45004	1	500	46.31
85/86	loy	1500	6160	7547.0	0	0.0	87.0	569	66827	1	500	46.88
86/87	loy	2000	9092	10857.9	0	0.0	146.1	616	96607	2	500	51.89
87/88	loy	2000	12307	13725.3	0	0.0	123.3	802	120763	2	500	70.25
88/89	loy	2000	13616	16859.0	0	0.0	87.0	-9	146910	3	500	77.72
89/90	loy	2000	14920	18028.0	0	0.0	60.0	695	156367	4	500	85.16

90/91	loy	2000	14873	18030.0	109	0.0	47.5	534	156217	5	500	84.89
81/82	haz	1600	8174	11646.8	0	0.0	216.3	1566	107272	14	200	58.32
82/83	haz	1600	6697	9566.3	0	0.0	146.4	1585	87419	15	200	47.78
83/84	haz	1600	6829	9368.4	0	0.0	136.8	1787	85465	16	200	48.72
84/85	haz	1600	7565	11031.3	0	0.0	80.5	1634	98854	17	200	53.97
85/86	haz	1600	6984	10034.6	0	0.0	74.8	1528	89958	18	200	49.83
86/87	haz	1600	7272	10354.3	0	0.0	81.1	1266	92910	19	200	51.88
87/88	haz	1600	7285	10306.0	0	0.0	42.7	1191	91636	20	200	51.98
88/89	haz	1600	8397	12361.0	0	0.0	63.4	-9	110178	21	200	59.91
89/90	haz	1600	6995	10468.0	0	0.0	76.0	1675	93798	22	200	49.91
90/91	haz	1600	8445	12435.6	9.45	0.0	38.6	1127	110296	23	200	60.25
81/82	yal	1450	7608	11816.0	0	90.3	0.0	988	91385	4	362.5	59.90
82/83	yal	1450	8882	14440.8	0	45.0	0.0	1059	108828	5	362.5	69.93
83/84	yal	1450	8248	13487.1	0	28.4	0.0	1103	101046	6	362.5	64.93
84/85	yal	1450	8375	13897.3	0	17.7	0.0	1096	103614	7	362.5	65.93
85/86	yal	1450	7551	11548.6	0	20.0	0.0	1084	86334	8	362.5	59.45
86/87	yal	1450	8460	13891.2	0	13.6	0.0	986	103389	9	362.5	66.60
87/88	yal	1450	8006	13140.6	0	13.7	0.0	901	97839	10	362.5	63.03
88/89	yal	1450	8570	14016.6	0	24.7	0.0	-9	104802	11	362.5	67.47
89/90	yal	1450	8463	13200.0	0	29.6	0.0	923	98974	12	362.5	66.63
90/91	yal	1450	9479	14690.1	0	17.0	0.0	648	109450	13	362.5	74.63
81/82	mor	170	1010	2092.9	0	1.2	0.0	476	18470	22	34	67.82
82/83	mor	170	1092	2115.0	0	1.3	0.0	467	18669	23	34	73.33
83/84	mor	170	1059	2073.8	0	1.2	0.0	452	18302	24	34	71.11
84/85	mor	170	868	1821.0	0	1.4	0.0	466	16086	25	34	58.29
85/86	mor	170	805	1810.2	0	1.8	0.0	435	16008	26	34	54.06
86/87	mor	170	762	1721.5	0	1.1	0.0	399	15197	27	34	51.17
87/88	mor	170	742	1757.6	0	1.3	0.0	396	15524	28	34	49.83
88/89	mor	170	835	1816.1	0	1.0	0.0	-9	16025	29	34	56.07
89/90	mor	170	531	1371.6	0	1.3	0.0	387	12127	30	34	35.66
90/91	mor	170	462	1203.6	0	1.2	0.0	344	10644	31	34	31.02
85/86	bay	1320	5562	2792.0	0	6.8	0.0	400	65645	0	660	48.10
86/87	bay	2640	11451	5175.0	0	8.5	0.0	519	121485	1	660	49.51
87/88	bay	2640	10288	4540.0	0	8.5	0.0	561	106626	1	660	44.49
88/89	bay	2640	12593	5640.0	0	6.7	0.0	599	132284	2	660	54.45
89/90	bay	2640	14182	6116.0	0	3.3	0.0	603	143266	3	660	61.32
90/91	bay	2640	15406	6603.0	0	3.5	0.0	556	154671	4	660	66.62
82/83	era	1320	5817	2526.0	0	6.5	0.0	390	60670	0	660	50.31

83/84	era	1980	8337	3586.0	0	9.7	0.0	518	86151	1	660	48.07
84/85	era	2640	10089	4580.0	0	10.6	0.0	560	109949	1	660	43.63
85/86	era	2640	8931	4147.0	0	10.1	0.0	641	99577	2	660	38.62
86/87	era	2640	9680	4542.0	0	8.4	0.0	672	108939	3	660	41.86
87/88	era	2640	10284	4568.0	0	8.4	0.0	669	109561	4	660	44.47
88/89	era	2640	11429	4980.0	0	8.0	0.0	694	119389	5	660	49.42
89/90	era	2640	11230	4908.0	0	6.3	0.0	644	117590	6	660	48.56
90/91	era	2640	12539	5279.0	0	4.0	0.0	583	126352	7	660	54.22
82/83	lid	2000	8551	4448.0	0	6.5	0.0	662	99047	10	500	48.81
83/84	lid	2000	9966	5096.0	0	13.4	0.0	683	113753	11	500	56.88
84/85	lid	2000	9980	5133.0	0	14.2	0.0	645	114611	12	500	56.96
85/86	lid	2000	6948	3601.0	0	18.0	0.0	653	80777	13	500	39.66
86/87	lid	2000	6214	3146.0	0	22.5	0.0	652	70885	14	500	35.47
87/88	lid	2000	6660	3433.0	0	13.9	0.0	645	76858	15	500	38.01
88/89	lid	2000	5673	2883.0	0	13.2	0.0	590	64615	16	500	32.38
89/90	lid	2000	6177	3150.0	0	10.9	0.0	571	70436	17	500	35.26
90/91	lid	2000	5190	2569.0	0	7.8	0.0	554	57394	18	500	29.62
82/83	mun	1200	5482	2538.0	0	5.8	0.0	562	67269	13	300	52.15
83/84	mun	1200	4927	2268.0	0	4.3	0.0	565	60073	14	300	46.87
84/85	mun	1200	3451	1622.0	0	4.5	0.0	570	43027	15	300	32.83
85/86	mun	1200	3937	1773.0	0	2.9	0.0	585	46940	16	300	37.45
86/87	mun	1200	3509	1582.0	0	4.3	0.0	614	41962	17	300	33.38
87/88	mun	1200	3094	1385.0	0	4.9	0.0	609	36789	18	300	29.43
88/89	mun	1200	3216	1502.0	0	13.1	0.0	576	40254	19	300	30.59
89/90	mun	1200	3125	1406.0	0	3.8	0.0	589	37293	20	300	29.73
90/91	mun	1200	2652	1507.0	0	4.6	0.0	452	39996	21	300	25.23
82/83	val	1320	7481	3173.0	0	9.3	0.0	849	75924	3	660	64.70
83/84	val	1320	8176	3254.0	0	6.1	0.0	873	77712	4	660	70.71
84/85	val	1320	7572	3176.0	0	11.5	0.0	854	76091	5	660	65.48
85/86	val	1320	6814	2907.0	0	11.3	0.0	884	69680	6	660	58.93
86/87	val	1320	5945	2618.0	0	8.3	0.0	852	62671	7	660	51.41
87/88	val	1320	6118	2745.0	0	13.6	0.0	800	65925	8	660	52.91
88/89	val	1320	5672	2828.0	0	13.7	0.0	696	67905	9	660	49.05
89/90	val	1320	6589	2777.0	0	4.6	0.0	449	66294	10	660	56.98
90/91	val	1320	6309	2662.0	0	4.6	0.0	417	63557	11	660	54.56
82/83	wal	1000	3271	1714.0	0	8.4	0.0	591	43236	4	500	37.34
83/84	wal	1000	2703	1297.0	0	4.3	0.0	596	32622	5	500	30.86
84/85	wal	1000	3545	1595.0	0	5.5	0.0	573	40127	6	500	40.47

85/86	wal	1000	5238	2311.0	0	3.9	0.0	586	57954	7	500	59.79
86/87	wal	1000	2458	1149.0	0	6.5	0.0	585	29023	8	500	28.06
87/88	wal	1000	4479	1888.0	0	4.7	0.0	593	47416	9	500	51.13
88/89	wal	1000	4171	1883.0	0	4.3	0.0	520	47272	10	500	47.61
89/90	wal	1000	4259	1901.0	0	3.8	0.0	496	47699	11	500	48.62
90/91	wal	1000	2750	1254.0	0	3.4	0.0	451	31506	12	500	31.39
81/82	bun	120	-9	323.8	0	1.9	0.0	182	6557	23	30	na
82/83	bun	120	401	282.5	0	2.4	0.0	179	5755	24	30	38.19
83/84	bun	120	374	269.9	0	2.6	0.0	171	5508	25	30	35.64
84/85	bun	120	155	115.3	0	2.4	0.0	195	2409	26	30	14.78
85/86	bun	120	99	80.3	0	1.5	0.0	144	1671	27	30	9.47
86/87	bun	120	94	74.2	0	1.5	0.0	138	1548	28	30	9.01
87/88	bun	120	65	60.6	0	1.6	0.0	132	1282	29	30	6.19
88/89	bun	120	177	136.9	0	1.8	0.0	134	2816	30	30	16.86
89/90	bun	120	165	127.4	0	1.5	0.0	136	2612	31	30	15.76
90/91	bun	120	129	103.3	0	1.4	0.0	136	2123	32	30	12.30
81/82	muja	640	2844	1764.4	0	6.5	0.0	464	35569	6	107	50.74
82/83	muja	640	3233	2000.1	0	5.7	0.0	525	40248	7	107	57.67
83/84	muja	640	3113	1919.8	0	6.5	0.0	562	38677	8	107	55.54
84/85	muja	840	3973	2353.1	0	6.2	0.0	579	47327	9	107	53.99
85/86	muja	1040	3723	2242.2	0	8.8	0.0	732	45221	7	120	40.87
86/87	muja	1040	3563	1981.7	0	11.5	0.0	700	40127	6	130	39.12
87/88	muja	1040	3698	2251.9	0	6.6	0.0	668	45323	7	130	40.60
88/89	muja	1040	4828	2865.5	0	10.0	0.0	704	57741	8	130	53.00
89/90	muja	1040	5549	3210.0	0	10.9	0.0	741	64667	9	130	60.92
90/91	muja	1040	6111	3471.9	0	9.2	0.0	736	69831	10	130	67.08
81/82	kwin	880	1502	684.7	0	73.6	0.0	377	16793	9	147	19.49
82/83	kwin	880	1336	704.1	0	61.5	0.0	382	16656	10	147	17.34
83/84	kwin	880	1698	940.2	537	27.3	0.0	424	20423	11	147	22.03
84/85	kwin	880	1817	852.7	3220	18.9	0.0	432	20999	12	147	23.58
85/86	kwin	880	2443	63.8	27060	8.5	0.0	456	28694	13	147	31.69
86/87	kwin	880	3025	112.9	33137	3.7	0.0	439	35544	14	147	39.25
87/88	kwin	880	3512	84.4	40610	1.1	0.0	458	42336	15	147	45.56
88/89	kwin	880	3115	230.5	32288	4.7	0.0	481	37075	16	147	40.41
89/90	kwin	880	3076	325.5	29457	6.3	0.0	473	36205	17	147	39.91
90/91	kwin	880	2913	723.2	20140	1.1	0.0	503	34581	18	147	37.80

Appendix 4: FRONTIER Programmer's Guide

The FRONT41.000 File

The start-up file FRONT41.000 is listed in Table A4.1. Ten values may be altered in FRONT41.000. A brief description of each value is provided below.

Table A4.1

The Start-up File for the FRONTIER Program: FRONT41.000

KEY VALUES USED IN FRONTIER PROGRAM (VERSION 4.1)

NUMBER:	DESCRIPTION:
5	IPRINT - PRINT INFO EVERY "N" ITERATIONS, 0=DO NOT PRINT
1	INDIC - USED IN UNIDIMENSIONAL SEARCH PROCEDURE - SEE BELOW
0.00001	TOL - CONVERGENCE TOLERANCE (PROPORTIONAL)
0.001	TOL2 - TOLERANCE USED IN UNI-DIMENSIONAL SEARCH PROCEDURE
1.0D+16	BIGNUM - USED TO SET BOUNDS ON DEN & DIST
0.00001	STEP1 - SIZE OF 1ST STEP IN SEARCH PROCEDURE
1	IGRID2 - 1=DOUBLE ACCURACY GRID SEARCH, 0=SINGLE
0.1	GRIDNO - STEPS TAKEN IN SINGLE ACCURACY GRID SEARCH ON GAMMA
100	MAXIT - MAXIMUM NUMBER OF ITERATIONS PERMITTED
1	ITE - 1=PRINT ALL TE ESTIMATES, 0=PRINT ONLY MEAN TE

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.
FOR MORE INFORMATION ON THESE VARIABLES SEE WORKING PAPER: COELLI (1994) FROM THE UNIVERSITY OF NEW ENGLAND, ARMIDALE, NSW, 2351, AUSTRALIA.
INDIC VALUES:
indic=2 says do not scale step length in unidimensional search
indic=1 says scale (to length of last step) only if last step was smaller
indic= any other number says scale (to length of last step)

---

1) IPRINT - specifies how often information on the likelihood function value and the vector of parameter estimates should be recorded during the iterative process. It is initially set to 5, hence information is printed every 5 iterations. It can be set to

any non-negative integer value. A 0 will result in no reporting of intermediate information.

2) INDIC - 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 says do not scale step length in uni-dimensional search; indic=1 says scale (to length of last step) only if last step was smaller; and indic=any other number says scale (to length of last step) For more information see Himmelblau (1972).

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

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

5) BIGNUM - bounds the size of the largest 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 (which evaluates the standard normal density function) and DIS (which evaluates the standard normal distribution function). Errors with numerical underflows and overflows were the problems most frequently encountered by people attempting to install earlier versions of this program on various mainframe computers. This number has been set to 1.0e+16 for the IBM PC. 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 may be gained if larger numbers are permitted.

6) STEP1 - sets 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 sensible parameter space.

7) IGRID2 - a flag which if set to 1 will cause the grid search to complete a second phase grid search around the estimate obtained in the first phase of the grid search.

If set to zero 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 - sets the width of the steps taken in the grid search between zero and one on the γ parameter. For more information refer to the description of the grid search in Section 3.

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

10) ITE - specifies whether individual efficiency estimates should be listed in the output file. A value of 1 will cause them to be listed, and a 0 will suppress them.

Subroutine and Function Descriptions

SUBROUTINES:

1) EXEC: This is the main calling program. It firstly reads the start-up file FRONT2.000 before calling INFO.

2) INFO: This subroutine reads instructions either from a file or from the terminal then reads the data file. It then calls MINI.

3) 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.

4) 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.

5) GRID: Does a grid search over γ.

6) SEARCH: Performs a uni-dimensional search to determine the optimal step length for the next iteration. The Coggin method is used (see Himmelblau, 1972).

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

8) CONVRG: Tests the convergence critereon 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.

9) FUN1: Calculates the negative of the log-likelihood function (LLF) of Model 1. Note that FRONTIER minimizes the negative of the LLF which is equivalent to maximizing the LLF.

10) DER1: Calculates the first partial derivatives of the negative of the LLF of Model 1.

11) FUN2: Calculates the negative of the log-likelihood function (LLF) of Model 2.

12) DER2: Calculates the first partial derivatives of the negative of the LLF of Model 2.

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

14) 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.

15) INVERT: Inverts a given matrix.

FUNCTIONS:

1) DEN: Evaluates the density function of a standard normal random variable.

2) DIS: Evaluates the distribution function of a standard normal random variable.

Appendix 5: FRONTIER Program Listing

```
      program exec
c     FRONTIER version 4.1 by Tim Coelli.
c     This program uses the Davidon-Fletcher-Powell algorithm to
c     estimate two forms of the stochastic frontier production function.
c     The first is the error components model described in Battese and
c     Coelli (1992) in the JPA, and the second is the TE effects model
c     described in Battese and Coelli (1993), UNE Working Paper #69.
c     A large proportion of the search, convrg, mini and eta
c     subroutines are taken from the book: Himmelblau (1972, 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 charge.
c     If you find the program useful, a contribution of $150 US
c     to help defray some of the author's costs would be appreciated, but
c     is in no way obligatory.
c     Please note that 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. He may be
c     contacted via email at tcoelli@metz.une.edu.au .
c     Postal address:  Tim Coelli, Department of Econometrics,
c     University of New England, Armidale, NSW 2351, Australia,
c     last update = 17/October/1994
      implicit double precision (a-h,o-z)
      character*12 koutf,kdatf,kinf
      common/eight/koutf,kdatf,kinf
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      open(unit=70,file='front41.000',status='old')
      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,*) maxit
      read(70,*) ite
      close(70)
      nfunct=0
      ndrv=0
      call info
      stop
      end

      subroutine mini(yy,xx,mm,sv)
c     contains the main loop of this iterative program.
      implicit double precision (a-h,o-z)
      character*12 koutf,kdatf,kinf
      common/eight/koutf,kdatf,kinf
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
```

```
      dimension yy(nn,nt),xx(nn,nt,nr),mm(nn),sv(n)
      dimension ob(:),gb(:),obse(:),x(:),y(:),s(:)
      dimension h(:,:),delx(:),delg(:),gx(:),gy(:)
      allocatable :: ob,gb,obse,x,y,s,h,delx,delg,gx,gy
      allocate(ob(n),gb(n),obse(n),x(n),y(n),s(n))
      allocate(h(n,n),delx(n),delg(n),gx(n),gy(n))
      open(unit=70,file=koutf,status='unknown')
      do 98 i=1,n
      gx(i)=0.0
      gy(i)=0.0
98    continue
      call ols(ob,obse,yy,xx)
      if (igrid.eq.1) then
      write(6,*) 'doing grid search...'
      call grid(x,y,yy,xx,ob,gb)
      else
      do 131 i=1,n
      y(i)=sv(i)
      x(i)=sv(i)
131   continue
      if (im.eq.1) call fun1(x,fx,yy,xx)
      if (im.eq.2) call fun2(x,fx,yy,xx)
      fy=fx
      end if
      call result(yy,xx,mm,h,y,sv,ob,obse,gb,1)
      write(6,*) 'DFP iterative process underway - please wait...'
      iter=0
      if (im.eq.1) call der1(x,gx,yy,xx)
      if (im.eq.2) call der2(x,gx,yy,xx)
      write(70,*)
      write(70,*)
      write(70,301) iter,nfunct,-fy
      nc=1
305   write(70,302) (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.ne.0) write(70,2100)
2100   format(' gradient step')
      do 30 i=1,n
30    s(i)=-gx(i)
40    call search(x,y,s,gx,delx,yy,xx)
      iter=iter+1
      if (iter.ge.maxit) then
      write(70,*) 'maximum number of iterations reached'
      goto 70
      endif
7     if(fy.gt.fx) goto 5
      if (im.eq.1) call der1(y,gy,yy,xx)
      if (im.eq.2) call der2(y,gy,yy,xx)
      call convrg(ipass,x,y)
      if (ipass.eq.1.) goto 70
      if (iprint.ne.0) then
      printcon=float(iter)/float(iprint)-float(iter/iprint)
```

```
      if (printcon.eq.0.0) then
      write(70,301) iter,nfunct,-fy
      nc=1
304   write(70,302) (y(i),i=nc,min(n,nc+4))
      nc=nc+5
      if(nc.le.n) goto 304
      endif
      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(h,delx,delg,gx)
      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(70,301) iter,nfunct,-fy
      nc=1
303   write(70,302) (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)
      call result(yy,xx,mm,h,y,sv,ob,obse,gb,2)
      deallocate(ob,gb,obse,x,y,s,h,delx,delg,gx,gy)
      close(70)
      return
      end

      subroutine convrg(ipass,x,y)
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)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      dimension x(n),y(n)
      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(h,delx,delg,gx)
c    calculates the direction matrix (p).
     implicit double precision (a-h,o-z)
     common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
     common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
     dimension h(n,n),delx(n),delg(n),gx(n)
     dimension hdg(:),dgh(:),hgx(:)
     allocatable :: hdg,dgh,hgx
     allocate(hdg(n),dgh(n),hgx(n))
     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.1.0/bignum) 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
     deallocate(hdg,dgh,hgx)
     return
     end

     subroutine search(x,y,s,gx,delx,yy,xx)
c    unidimensional search (coggin) to determine optimal step length
c    determines the step length (t) using a unidimensional search.
```

```
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      dimension x(n),y(n),s(n),gx(n),delx(n)
      dimension yy(nn,nt),xx(nn,nt,nr)
      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)
      if (im.eq.1) call fun1(y,f,yy,xx)
      if (im.eq.2) call fun2(y,f,yy,xx)
      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.ne.0) write(70,2100)
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)
      if (im.eq.1) call fun1(y,f,yy,xx)
      if (im.eq.2) call fun2(y,f,yy,xx)
 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.ne.0) write(70,2500)
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)
      if (im.eq.1) call fun1(y,f,yy,xx)
      if (im.eq.2) call fun2(y,f,yy,xx)
      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(70,3000) 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(70,5000)
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.ne.0) write(70,2000)
2000  format(' pt better than entering pt cannot be found')
      return
      end

      subroutine check(b,xx)
c     checks if params are out of bounds & adjusts if required.
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      dimension b(n),xx(nn,nt,nr)
      n1=nr+1
      n2=nr+2
      bi=dsqrt(bignum)
      if(b(n1).le.0.0) b(n1)=1.0/bi
      if(b(n2).le.1.0/bi) b(n2)=1.0/bi
      if(b(n2).ge.1.0-1.0/bi) b(n2)=1.0-1.0/bi
      bound=2.*dsqrt(b(n1)*b(n2))
      if((im.eq.1).and.(nmu.eq.1)) then
      n3=nr+3
      if(b(n3).gt.bound) b(n3)=bound
      if(b(n3).lt.-bound) b(n3)=-bound
      endif
      return
      end

      subroutine fun1(b,a,yy,xx)
c     calculates the negative of the log-likelihood function of the
c     error components model.
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      data pi/3.1415926/
      dimension b(n),yy(nn,nt),xx(nn,nt,nr)
      call check(b,xx)
      a=0.0
```

```
      f=dfloat(nn)
      fnt=dfloat(nt)
      ftot=dfloat(nob)
      s2=b(nb+1)
      g=b(nb+2)
      u=0.0
      e=0.0
      if (nmu.eq.1) then
      u=b(nb+3)
      if (neta.eq.1) e=b(nb+4)
      else
      if (neta.eq.1) e=b(nb+3)
      endif
      sc=1.
      if (ipc.eq.2) sc=-1.
      a=0.5*ftot*(dlog(2.0*pi)+dlog(s2))
      a=a+0.5*(ftot-f)*dlog(1.0-g)
      z=u/(s2*g)**0.5
      a=a+f*dlog(dis(z))
      a=a+0.5*f*z**2
      a2=0.0
      do 132 i=1,nn
      epr=0.0
      do 103 l=1,nt
      if (xx(i,l,1).ne.0.0) then
      ee=yy(i,l)
      do 102 j=1,nb
      ee=ee-b(j)*xx(i,l,j)
 102  continue
      epr=epr+ee*dexp(-e*(dfloat(l)-fnt))
      end if
 103  continue
      epe=0.0
      do 101 l=1,nt
      if (xx(i,l,1).ne.0.0) epe=epe+dexp(-2.0*e*(dfloat(l)-fnt))
 101  continue
      zi=(u*(1.0-g)-sc*g*epr)/(g*(1.0-g)*s2*(1.0+(epe-1.0)*g))**0.5
      a=a+0.5*dlog(1.0+(epe-1.0)*g)
      a=a-dlog(dis(zi))
      do 133 l=1,nt
      if (xx(i,l,1).ne.0.0) then
      ee=yy(i,l)
      do 134 j=1,nb
      ee=ee-b(j)*xx(i,l,j)
 134  continue
      a2=a2+ee**2
      end if
 133  continue
      a=a-0.5*zi**2
 132  continue
      a=a+0.5*a2/((1.0-g)*s2)
      nfunct=nfunct+1
      return
      end

      subroutine der1(b,gx,yy,xx)
c     calculates the first-order partial derivatives of the negative
c     of the log-likelihood function of the error components model.
```

```
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      dimension b(n),gx(n),yy(nn,nt),xx(nn,nt,nr)
      call check(b,xx)
      f=dfloat(nn)
      ftot=dfloat(nob)
      fnt=dfloat(nt)
      n1=nr+1
      n2=nr+2
      s2=b(n1)
      g=b(n2)
      u=0.0
      e=0.0
      if (nmu.eq.1) then
      n3=nr+3
      u=b(n3)
      if (neta.eq.1) then
      n4=nr+4
      e=b(n4)
      endif
      else
      if (neta.eq.1) then
      n4=nr+3
      e=b(n4)
      endif
      endif
      sc=1.
      if (ipc.eq.2) sc=-1.
      z=u/(s2*g)**0.5
      do 106 j=1,n
      gx(j)=0.0
106   continue
      gx(n1)=0.5*ftot/s2-0.5*f*(den(-z)/(dis(z))+z)*z/s2
      gx(n2)=-.5*(ftot-f)/(1.-g)-.5*f*(den(-z)/(1.-dis(-z))+z)*z/g

      do 105 i=1,nn
      epr=0.0
      epe=0.0
      do 103 l=1,nt
      if (xx(i,l,1).ne.0.0) then
      ee=yy(i,l)
      do 102 j=1,nb
      ee=ee-b(j)*xx(i,l,j)
102   continue
      epr=epr+ee*dexp(-e*(dfloat(l)-fnt))
      epe=epe+dexp(-2.0*e*(dfloat(l)-fnt))
      end if
103   continue
      zi=(u*(1.0-g)-sc*g*epr)/(g*(1.0-g)*s2*(1.0+(epe-1.0)*g))**0.5

      do 132 j=1,nb
      do 134 l=1,nt
      if(xx(i,l,1).ne.0.0) then
      ee=yy(i,l)
      do 135 k=1,nb
      ee=ee-xx(i,l,k)*b(k)
135   continue
```

```
      gx(j)=gx(j)-xx(i,l,j)*ee/(s2*(1.-g))
      endif
134   continue
      xpe=0.0
      do 146 l=1,nt
      if(xx(i,l,1).ne.0.0)xpe=xpe+xx(i,l,j)*dexp(-e*(dfloat(l)-fnt))
146   continue
      d=(den(-zi)/(dis(zi))+zi)*g*xpe*sc
      gx(j)=gx(j)-d/(g*(1.0-g)*s2*(1.0+(epe-1.0)*g))**0.5
132   continue

      gx(n1)=gx(n1)+.5*(den(-zi)/(dis(zi))+zi)*zi/s2
      ss=0.0
      do 138 l=1,nt
      ee=yy(i,l)
      do 139 j=1,nb
      ee=ee-xx(i,l,j)*b(j)
139   continue
      ss=ss+ee**2
138   continue
      gx(n1)=gx(n1)-0.5*ss/((1.0-g)*s2**2)

      gx(n2)=gx(n2)+0.5*ss/((1.0-g)**2*s2)
      gx(n2)=gx(n2)+0.5*(epe-1.0)/(1.0+(epe-1.0)*g)
      d=g*(1.-g)*(1.0+(epe-1.0)*g)
      dzi=-(u+sc*epr)*d
      c=0.5*(u*(1.0-g)-sc*g*epr)
      dzi=dzi-c*((1.0-2.0*g)+(epe-1.0)*g*(2.0-3.0*g))
      dzi=dzi/(d**1.5*s2**0.5)
      gx(n2)=gx(n2)-(den(-zi)/(dis(zi))+zi)*dzi

      if (nmu.eq.1) then
      gx(n3)=gx(n3)+1./(s2*g)**0.5*(den(-z)/(dis(z))+z)
      d=(den(-zi)/(dis(zi))+zi)*(1.-g)
      gx(n3)=gx(n3)-d/(g*(1.-g)*s2*(1.+(epe-1.)*g))**.5
      end if

      if (neta.eq.1) then
      de=0.0
      d=0.0
      do 152 l=1,nt
      if (xx(i,l,1).eq.1) then
      t=dfloat(l)
      de=de-2.0*(t-fnt)*dexp(-2.0*e*(t-fnt))
      ee=yy(i,l)
      do 153 j=1,nb
      ee=ee-xx(i,l,j)*b(j)
153   continue
      d=d+(t-fnt)*dexp(-e*(t-fnt))*ee
      end if
152   continue
      dd=(g*(1.0-g)*s2*(1.0+(epe-1.0)*g))
      d=d*g*dd*sc
      c=u*(1.0-g)-sc*g*epr
      c=c*0.5*g**2*(1.0-g)*s2*de
      dzi=(d-c)/dd**1.5
      gx(n4)=gx(n4)-(den(-zi)/(dis(zi))+zi)*dzi
      gx(n4)=gx(n4)+g/2.0*de/(1.0+(epe-1.0)*g)
```

```
      end if
105   continue

      ndrv=ndrv+1
      return
      end


      subroutine fun2(b,a,yy,xx)
c     calculates the negative of the log-likelihood function of the
c     TE effects model.
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      dimension b(n),yy(nn,nt),xx(nn,nt,nr)
      data pi/3.1415926/
      call check(b,xx)
      s2=b(nr+1)
      g=b(nr+2)
      ss=(g*(1.-g)*s2)**0.5
      sc=1.0
      if (ipc.eq.2) sc=-1.0
      a=0.
      do 10 i=1,nn
      do 10 l=1,nt
      if (xx(i,l,1).ne.0.0) then
      xb=0.
      do 11 j=1,nb
      xb=xb+xx(i,l,j)*b(j)
11    continue
      ee=(yy(i,l)-xb)
      zd=0.
      if (nz.ne.0) then
      do 12 j=nb+1,nr
      zd=zd+xx(i,l,j)*b(j)
12    continue
      endif
      us=(1.-g)*zd-sc*g*ee
      d=zd/(g*s2)**0.5
      ds=us/ss
      a=a-0.5*dlog(2.*pi)-0.5*dlog(s2)-(dlog(dis(d))-dlog(dis(ds)))
     + -0.5*(ee+sc*zd)**2/s2
      endif
10    continue
      a=-a
      nfunct=nfunct+1
      return
      end

      subroutine der2(b,gx,yy,xx)
c     calculates the first-order partial derivatives of the negative
c     of the log-likelihood function of the TE effects model.
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      dimension b(n),gx(n),yy(nn,nt),xx(nn,nt,nr)
      call check(b,xx)
      s2=b(nr+1)
```

```
      g=b(nr+2)
      ss=(g*(1.-g)*s2)**0.5
      sc=1.0
      if (ipc.eq.2) sc=-1.0
      do 9 j=1,n
      gx(j)=0.
 9    continue
      do 10 i=1,nn
      do 10 l=1,nt
      if (xx(i,l,1).ne.0.0) then
      xb=0.
      do 11 j=1,nb
      xb=xb+xx(i,l,j)*b(j)
 11   continue
      ee=(yy(i,l)-xb)
      zd=0.
      if (nz.ne.0) then
      do 12 j=nb+1,nr
      zd=zd+xx(i,l,j)*b(j)
 12   continue
      endif
      us=(1.-g)*zd-sc*g*ee
      d=zd/(g*s2)**0.5
      ds=us/ss
      do 13 j=1,nb
      gx(j)=gx(j)+xx(i,l,j)*((ee+sc*zd)/s2+sc*den(ds)/dis(ds)*g/ss)
 13   continue
      if (nz.ne.0) then
      do 14 j=nb+1,nr
      gx(j)=gx(j)-xx(i,l,j)*((sc*ee+zd)/s2+den(d)/dis(d)/(g*s2)
     + **0.5-den(ds)/dis(ds)*(1.-g)/ss)
 14   continue
      endif
      gx(nr+1)=gx(nr+1)-0.5/s2*(1.-(den(d)/dis(d)*d-den(ds)
     + /dis(ds)*ds)-(ee+sc*zd)**2/s2)
      gx(nr+2)=gx(nr+2)+0.5*(den(d)/dis(d)*d/g-den(ds)/dis(ds)
     + /ss*(zd/g+sc*ee/(1.-g)))
c     gx(nr+2)=gx(nr+2)+0.5*(den(d)/dis(d)*d/g-den(ds)/dis(ds)*
c    + (2.*(ee+zd)/ss+ds*(1.-2.*g)/(g*(1.-g))))
      endif
 10   continue
      do 15 j=1,n
      gx(j)=-gx(j)
 15   continue
      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,gridno,igrid2,ite
      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,gridno,igrid2,ite
      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.1.0/bignum) 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

      subroutine info
c     accepts instructions from the terminal or from a file and
c     also reads data from a file.
      implicit double precision (a-h,o-z)
      character*12 koutf,kdatf,kinf
      common/eight/koutf,kdatf,kinf
```

```
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      character chop,chst,chmu,cheta,chl
      dimension yy(:,:),xx(:,:,:),mm(:),sv(:),xxd(:)
      allocatable :: yy,xx,mm,sv,xxd
      nmu=0
      neta=0
      igrid=1
      nz=0
      kinf='terminal'
      write(6,*)
      write(6,*)
      write(6,*) '              **********************'
      write(6,*) '              FRONTIER - Version 4.1'
      write(6,*) '              **********************'
      write(6,*)
      write(6,*) '                      by'
      write(6,*) '                  Tim Coelli'
      write(6,*) '            Department of Econometrics'
      write(6,*) '             University of New England'
      write(6,*) '               Armidale, NSW, 2351'
      write(6,*) '                  Australia.'
      write(6,*) '          (email: tcoelli@metz.une.edu.au)'
      write(6,*)
      write(6,*) '[This software is not a commercial product.'
      write(6,*) 'If you find it useful, a contribution of'
      write(6,*) 'A$200 (US$150) to help cover some of the costs'
      write(6,*) 'associated with development would be'
      write(6,*) 'appreciated - but is in no way obligatory.]'
      write(6,*)
      write(6,*)
      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 1 if you wish to estimate the error '
      write(6,*) 'components model, or 2 for the TE effects model : '
      read(5,*) im
      if ((im.ne.1).and.(im.ne.2)) then
      write(6,*) ' incorrect option, must be 1 or 2 - bye!'
      stop
      endif
      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,*) ' are you estimating a production or cost function?'
      write(6,*) ' enter a 1 for production or a 2 for cost : '
      read(5,*) ipc
      write(6,*) ' is the dependent variable logged? (y or n) '
      read(5,61) chl
      if ((chl.eq.'y').or.(chl.eq.'Y')) il=1
      write(6,*) ' how many cross-sections in the data ? '
      read(5,*) nn
      write(6,*) ' how many time-periods in the data ? '
      read(5,*) nt
      write(6,*) ' how many observations in total in the data ? '
```

```
      read(5,*) nob
      if ((nn*nt).lt.nob) 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 regressors (Xs) are there ? '
      read(5,*) nb
      if (im.eq.1) then
      write(6,*) ' does the model include mu ? (y or n)  '
      read(5,61) chmu
      if ((chmu.eq.'y').or.(chmu.eq.'Y')) nmu=1
      if (nt.gt.1) then
      write(6,*) ' does the model include eta ? (y or n)  '
      read(5,61) cheta
      if ((cheta.eq.'y').or.(cheta.eq.'Y')) neta=1
      endif
      nb=nb+1
      nr=nb
      n=nr+2+nmu+neta
      else
      write(6,*)' does the model include delta0 ? (y or n)  '
      read(5,61) chmu
      if ((chmu.eq.'y').or.(chmu.eq.'Y')) nmu=1
      write(6,*)' how many eff.-term regressors (Zs) are there ? '
      read(5,*) nz
      nz=nz+nmu
      nb=nb+1
      nr=nb+nz
      n=nr+2
      endif
      allocate(sv(n))
      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(5,61) 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
      write(6,62) i-1
      read(5,*) sv(i)
138   continue
      write(6,*) ' enter starting value for sigma squared :  '
      read(5,*) sv(nr+1)
      write(6,*) ' enter starting value for gamma :  '
      read(5,*) sv(nr+2)
      if ((im.eq.2).and.(nz.gt.0)) then
63    format(' enter starting value for delta',i2,' : ')
      do 139 i=1,nz
      write(6,63) i-nmu
      read(5,*) sv(nb+i)
139   continue
      endif
      if (im.eq.1) then
      if (nmu.eq.1) then
      write(6,*) ' enter starting value for mu :  '
```

```
      read(5,*) sv(nb+3)
      if (neta.eq.1) then
      write(6,*) ' enter starting value for eta :  '
      read(5,*) sv(nb+4)
      end if
      else
      if (neta.eq.1) then
      write(6,*) ' enter starting value for eta :  '
      read(5,*) sv(nb+3)
      end if
      end if
      endif
      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,status='old')
      read(50,*) im
      read(50,60) kdatf
      read(50,60) koutf
      read(50,*) ipc
      read(50,61) chl
      if ((chl.eq.'y').or.(chl.eq.'Y')) il=1
      read(50,*) nn
      read(50,*) nt
      read(50,*) nob
      if ((nn*nt).lt.nob) 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
      if (im.eq.1) then
      read(50,61) chmu
      read(50,61) cheta
      if ((chmu.eq.'y').or.(chmu.eq.'Y')) nmu=1
      if ((cheta.eq.'y').or.(cheta.eq.'Y')) neta=1
      nb=nb+1
      nr=nb
      n=nr+2+nmu+neta
      else
      read(50,61) chmu
      if ((chmu.eq.'y').or.(chmu.eq.'Y')) nmu=1
      read(50,*) nz
      nz=nz+nmu
      nb=nb+1
      nr=nb+nz
      n=nr+2
      endif
      allocate (sv(n))
      read(50,61) chst
      if ((chst.eq.'y').or.(chst.eq.'Y')) then
      igrid=0
      if (im.eq.1) then
      do 148 i=1,n
      read(50,*) sv(i)
148   continue
```

```
      else
      do 152 i=1,nb
      read(50,*) sv(i)
152   continue
      read(50,*) sv(n-1)
      read(50,*) sv(n)
      if (nz.gt.0) then
      do 153 i=1,nz
      read(50,*) sv(nb+i)
153   continue
      endif
      endif
      end if
      else
      write(6,*) ' incorrect option - must be t or f - bye!'
      stop
      endif
      allocate(yy(nn,nt),xx(nn,nt,nr),mm(nn),xxd(nr))
      open(unit=40,file=kdatf,status='old')
      do 135 i=1,nn
      mm(i)=0
      do 135 l=1,nt
      xx(i,l,1)=0.0
135   continue
      do 134 k=1,nob
      ndat=nr
      if(im.eq.2) ndat=nr-nmu
      read(40,*) fii,ftt,yyd,(xxd(j),j=2,ndat)
      i=int(fii)
      l=int(ftt)
      mm(i)=mm(i)+1
      xx(i,l,1)=1.0
      yy(i,l)=yyd
      do 136 j=2,nb
      xx(i,l,j)=xxd(j)
136   continue
      if ((im.eq.2).and.(nz.gt.0)) then
      if (nmu.eq.1) xx(i,l,nb+1)=1.0
      if ((nz-nmu).gt.0) then
      do 154 j=nb+nmu+1,nr
      xx(i,l,j)=xxd(j-nmu)
154   continue
      endif
      endif
      if (i.lt.1) then
      write(6,*) ' error - a firm number is < 1'
      stop
      else if (i.gt.nn) then
      write(6,*) ' error - a firm number is > number of firms'
      stop
      else if (l.lt.1) then
      write(6,*) ' error - a period number is < 1'
      stop
      else if (l.gt.nt) then
      write(6,*) ' error - a period number is > number of periods'
      stop
      end if
134   continue
```

```
      do 149 i=1,nn
      if (mm(i).eq.0) then
      write(6,66) i
 66   format(' error - there are no observations on firm ',i6)
      stop
      end if
 149  continue
      call mini(yy,xx,mm,sv)
      deallocate(yy,xx,mm,sv,xxd)
      return
      end


      subroutine result(yy,xx,mm,h,y,sv,ob,obse,gb,ncall)
c     presents estimates, covariance matrix, standard errors and t-ratios,
c     as well as presenting many results including estimates of technical
c     efficiency.
      implicit double precision (a-h,o-z)
      character*12 koutf,kdatf,kinf
      common/eight/koutf,kdatf,kinf
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      dimension yy(nn,nt),xx(nn,nt,nr),mm(nn)
      dimension h(n,n),y(n),sv(n),ob(n),obse(n),gb(n)
      dimension mt(:)
      allocatable :: mt
      data pi/3.1415926/
      allocate(mt(nt))
      open (unit=70,file=koutf,status='unknown')
      n1=nr+1
      n2=nr+2
      n3=nr+3
      n4=nr+4
      if((nmu.eq.0).and.(neta.eq.1)) n4=nr+3
      if (ncall.eq.1) then
      write (70,401)
 401  format(/,'Output from the program FRONTIER (Version 4.1)',//)
      write (70,601) kinf,kdatf
 601  format('instruction file = ',a12,/,'data file = ',7x,a12,//)
      if (im.eq.1) then
      write(70,*) 'Error Components Frontier (see B&C 1992)'
      else
      write(70,*) 'Tech. Eff. Effects Frontier (see B&C 1993)'
      endif
      if (ipc.eq.1) then
      write(70,*) 'The model is a production function'
      else
      write(70,*) 'The model is a cost function'
      endif
      if (il.eq.1) then
      write(70,*) 'The dependent variable is logged'
      else
      write(70,*) 'The dependent variable is not logged'
      endif
      fnob=dfloat(nob)
      fnb=dfloat(nb)
      os2=ob(nb+1)*(fnob-fnb)/fnob
```

```
      fxols=fnob/2.0*(dlog(2.0*pi)+dlog(os2)+1.0)
      if(igrid.eq.1) then
      write (70,403)
403   format(//'the ols estimates are :',/)
      write (70,404)
404   format(17x,'coefficient    standard-error    t-ratio',/)
      do 132 i=1,nb
      write (70,302) i-1,ob(i),obse(i),ob(i)/obse(i)
132   continue
      write (70,203) ob(nb+1)
      write(70,501) -fxols
      endif
      if(igrid.eq.0) then
      write (70,402)
402   format(//'the starting values supplied were : ',/)
202   format(' beta',i2,7x,e16.8)
 82   format(' delta',i2,6x,e16.8)
203   format(' sigma-squared',e16.8)
204   format(' gamma',8x,e16.8)
205   format(' mu          ',e16.8)
206   format(' eta         ',e16.8)
      do 131 i=1,nb
      write (70,202) i-1,sv(i)
131   continue
      write (70,203) sv(n1)
      write (70,204) sv(n2)
      if ((nz.ne.0).and.(im.eq.2)) then
      do 83 i=1,nz
      write(70,82) i-nmu,sv(nb+i)
 83   continue
      endif
      if (im.eq.1) then
      if (nmu.eq.1) then
      write (70,205) sv(n3)
      else
      write (70,*) ' mu is restricted to be zero'
      end if
      if (neta.eq.1) then
      write (70,206) sv(n4)
      else
      write (70,*) ' eta is restricted to be zero'
      end if
      endif
      end if
      if(igrid.eq.1) then
      write (70,405)
405   format(/,'the estimates after the grid search were :',/)
      do 133 i=1,nb
      write (70,202) i-1,gb(i)
133   continue
      write (70,203) gb(n1)
      write (70,204) gb(n2)
      if ((nz.ne.0).and.(im.eq.2)) then
      do 84 i=1,nz
      write(70,82) i-nmu,gb(nb+i)
 84   continue
      endif
      if (im.eq.1) then
```

```
      if (nmu.eq.1) then
      write (70,205) gb(n3)
      else
      write (70,*) '  mu is restricted to be zero'
      end if
      if (neta.eq.1) then
      write (70,206) gb(n4)
      else
      write (70,*) '  eta is restricted to be zero'
      end if
      endif
      end if

      else

      write(6,151) koutf
151   format(' sending output to: ',a12)
      write (70,406)
406   format(//,'the final mle estimates are :',/)
      write (70,404)
302   format('  beta',i2,7x,3e16.8)
 81   format('  delta',i2,6x,3e16.8)
303   format('  sigma-squared',3e16.8)
304   format('  gamma',8x,3e16.8)
305   format('  mu          ',3e16.8)
306   format('  eta         ',3e16.8)
      do 134 i=1,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 ((nz.ne.0).and.(im.eq.2)) then
      do 199 i=nb+1,nb+nz
      write (70,81) i-nmu-nb,y(i),h(i,i)**0.5,y(i)/h(i,i)**0.5
199   continue
      endif
      if (im.eq.1) then
      if (nmu.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 (neta.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
      endif
      write (70,501) -fx
501   format(/,'log likelihood function = ',e16.8)
      if((fx-fxols).gt.0) then
      write(70,422)
422   format(/,'the likelihood value is less than that obtained',
  +  /,'using ols! - try again using different starting values')
      else
      chi=2.0*dabs(fx-fxols)
      write(70,5011) chi
5011  format(/,'LR test of the one-sided error = ',e16.8)
```

```
      if (im.eq.1) idf=nmu+neta+1
      if (im.eq.2) idf=nz+1
      write(70,5012) idf
5012  format('with number of restrictions = ',i1)
      write(70,*) '[note that this statistic has a mixed chi-square'
  +  ,' distribution]'
      end if
      write (70,502) iter
502   format(/,'number of iterations = ', i6)
      write(70,420) maxit
420   format(/,'(maximum number of iterations set at :',i6,')')
      write(70,513) nn
513   format(/,'number of cross-sections = ',i6)
      write(70,514) nt
514   format(/,'number of time periods = ',i6)
      write(70,515) nob
515   format(/,'total number of observations = ',i6)
      write(70,516) nn*nt-nob
516   format(/,'thus there are: ',i6,'  obsns not in the panel')

      write (70,58)
58   format(//,'covariance matrix :',/)
52   format(5e16.8)
      do 135 i=1,n
      nc=1
314   write(70,52) (h(i,j),j=nc,min(n,nc+4))
      nc=nc+5
      if(nc.le.n) goto 314
135   continue

      if((ite.eq.1).and.(ipc.eq.1)) write(70,503)
503   format(///,'technical efficiency estimates :',/)
      if((ite.eq.1).and.(ipc.eq.2)) write(70,504)
504   format(///,'cost efficiency estimates :',/)
67   format(//,1x,'mean efficiency = ',e16.8,////)
68   format(/,'    firm  year          eff.-est.',/)
69   format(2x,2i6,9x,e16.8)
167   format(//,1x,'mean eff. in year' ,i4,' =',e16.8,//)
168   format(/,'    firm          eff.-est.',/)
169   format(2x,i6,9x,e16.8)
      sc=1.
      if(ipc.eq.2) sc=-1.

      if (im.eq.1) then
      s2=y(nb+1)
      g=y(nb+2)
      u=0.0
      e=0.0
      if (nmu.eq.1) then
      u=y(nb+3)
      if (neta.eq.1) e=y(nb+4)
      else
      if (neta.eq.1) e=y(nb+3)
      endif
524   format(//,'efficiency estimates for year ',i6,' :')
505   format(2x,i6,4x,'no observation in this period')
      fnt=dfloat(nt)
      xbb=0.
```

```
      te=0.
      ncount=0
      ntt=nt
      if (neta.eq.0) ntt=1
      do 138 l=1,ntt
      t=dfloat(l)
      eta=dexp(-e*(t-fnt))
      if (ite.eq.1) then
      if (neta.eq.1) write(70,524) l
      write(70,168)
      endif
      do 136 i=1,nn
      if ((xx(i,l,1).ne.0.0).or.(neta.eq.0)) then
      epr=0.0
      epe=0.0
      do 103 k=1,nt
      if (xx(i,k,1).ne.0.0) then
      ee=yy(i,k)
      do 102 j=1,nb
      ee=ee-y(j)*xx(i,k,j)
 102  continue
      xb=yy(i,k)-ee
      xbb=xbb+xb
      epr=epr+ee*dexp(-e*(dfloat(k)-fnt))
      epe=epe+dexp(-2.0*e*(dfloat(k)-fnt))
      end if
 103  continue
      fi=(u*(1.0-g)-g*epr)/(1.0+(epe-1.0)*g)
      si2=g*(1.0-g)*s2/(1.0+(epe-1.0)*g)
      si=si2**0.5
      if (il.eq.1) then
      tei=(1.0-dis(sc*si*eta-fi/si))/(dis(fi/si))
      tei=tei*dexp(-fi*eta*sc+0.5*si2*eta**2)
      else
      tei=fi+si*den(fi/si)/dis(fi/si)
      tei=1.-sc*(eta*tei/(xb/dfloat(mm(i))))
      endif
      if ((ipc.eq.1).and.(tei.gt.1.0)) tei=1.0
      if ((ipc.eq.2).and.(tei.lt.1.0)) tei=1.0
      te=te+tei
      ncount=ncount+1
      endif
      if (ite.eq.1) then
      if ((xx(i,l,1).ne.0.).or.(neta.eq.0)) then
      write(70,169) i,tei
      else
      write(70,505) i
      endif
      endif
 136  continue
      if (neta.eq.1) write(70,167) l,te/dfloat(ncount)
      if (neta.eq.0) write(70,67) te/dfloat(ncount)
      te=0.
      ncount=0
 138  continue

      else
```

```
      if(ite.eq.1) write(70,68)
      s2=y(nr+1)
      g=y(nr+2)
      te=0.
      ss=(g*(1.-g)*s2)**0.5
      do 10 l=1,nt
      do 10 i=1,nn
      if (xx(i,l,1).ne.0.0) then
      xb=0.
      do 11 j=1,nb
      xb=xb+xx(i,l,j)*y(j)
 11   continue
      zd=0.
      if (nz.ne.0) then
      do 12 j=nb+1,nr
      zd=zd+xx(i,l,j)*y(j)
 12   continue
      endif
      us=(1.-g)*zd-g*(yy(i,l)-xb)
      ds=us/ss
      if (il.eq.1) then
      tei=dexp(-sc*us+0.5*ss**2)*dis(ds-sc*ss)/dis(ds)
      else
      tei=1.-sc*(us+ss*den(ds)/dis(ds))/xb
      endif
      if ((ipc.eq.1).and.(tei.gt.1.0)) tei=1.0
      if ((ipc.eq.2).and.(tei.lt.1.0)) tei=1.0
      if (ite.eq.1) write(70,69) i,l,tei
      te=te+tei
      endif
 10   continue
      te=te/dfloat(nob)
      write(70,67) te
      endif

      if ((nt.gt.1).and.(nt.le.100)) then
      write(70,441)
 441  format(///,'summary of panel of observations:',/,
     + '(1 = observed, 0 = not observed)',/)
      do 449 l=1,nt
      mt(l)=l
 449  continue
      write(70,442) (mt(l),l=1,nt)
 442  format(' t:',100i4)
      write(70,*) ' n'
 443  format(102i4)
      do 450 i=1,nn
      write(70,443) i,(int(xx(i,l,1)),l=1,nt),mm(i)
 450  continue
      do 451 l=1,nt
      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,nt),nob
 444  format(/,4x,101i4)
      write(70,445)
 445  format(////)
```

```
      endif
      endif
      deallocate(mt)
      return
      end



      subroutine grid(x,y,yy,xx,ob,gb)
c     does a grid search across gamma
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      dimension x(n),y(n),yy(nn,nt),xx(nn,nt,nr),ob(n),gb(n)
      data pi/3.1415926/
      n1=nr+1
      n2=nr+2
      sc=1.0
      if (ipc.eq.2) sc=-1.0
      var=ob(nb+1)*float(nob-nb)/float(nob)
      b0=ob(1)
      do 131 i=1,nb+1
      y(i)=ob(i)
 131  continue
      do 132 i=nb+1,nr
      y(i)=0.
 132  continue
      fx=bignum
      y6b=gridno
      y6t=1.0-gridno
      do 137 y6=y6b,y6t,gridno
      y(n2)=y6
      y(n1)=var/(1.-2.*y(n2)/pi)
      c=(y(n2)*y(n1)*2/pi)**0.5
      y(1)=b0+c*sc
      if (im.eq.1) call fun1(y,fy,yy,xx)
      if (im.eq.2) call fun2(y,fy,yy,xx)
      if(fy.lt.fx) then
      fx=fy
      do 138 i=1,n
      x(i)=y(i)
 138  continue
      end if
 137  continue
      if(igrid2.eq.1) then
      bb1=x(n2)-gridno/2.0
      bb2=x(n2)+gridno/2.0
      bb3=gridno/10.0
      do 140 y6=bb1,bb2,bb3
      y(n2)=y6
      y(n1)=var/(1.-2.*y(n2)/pi)
      c=(y(n2)*y(n1)*2/pi)**0.5
      y(1)=b0+c*sc
      if (im.eq.1) call fun1(y,fy,yy,xx)
      if (im.eq.2) call fun2(y,fy,yy,xx)
      if(fy.lt.fx) then
      fx=fy
```

```
      do 141 i=1,n
      x(i)=y(i)
141   continue
      end if
140   continue
      end if
      do 142 i=1,n
      gb(i)=x(i)
      y(i)=x(i)
142   continue
      fy=fx
      return
      end



      subroutine invert(xx,n)
c     finds the inverse of a given matrix.
      implicit double precision (a-h,o-z)
      common/five/tol,tol2,bignum,step1,gridno,igrid2,ite
      dimension xx(n,n)
      dimension ipiv(:)
      allocatable :: ipiv
      allocate(ipiv(n))
      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
      deallocate(ipiv)
      return
      end


      subroutine ols(ob,obse,yy,xx)
c     calculates the ols estimates and their standard errors.
```

```
      implicit double precision (a-h,o-z)
      common/one/fx,fy,nn,nz,nb,nr,nt,nob,nmu,neta,ipc,im,il
      common/three/n,nfunct,ndrv,iter,indic,iprint,igrid,maxit
      dimension ob(n),obse(n),yy(nn,nt),xx(nn,nt,nr)
      dimension xpx(:,:),xpy(:),mx(:)
      allocatable :: xpx,xpy,mx
      allocate(xpx(nb,nb),xpy(nb),mx(nb))
c     calculate x'x and x'y
      do 131 k=1,nb
      do 132 j=1,nb
      xpx(k,j)=0.0
      do 132 i=1,nn
      do 132 l=1,nt
      if (xx(i,l,1).ne.0.0) xpx(k,j)=xpx(k,j)+xx(i,l,k)*xx(i,l,j)
 132  continue
      xpy(k)=0.0
      do 131 i=1,nn
      do 131 l=1,nt
      if (xx(i,l,1).ne.0.0) xpy(k)=xpy(k)+xx(i,l,k)*yy(i,l)
 131  continue
c     determine correct scaling for x'x
      do 120 k=1,nb
      h=(1.0-dlog10(xpx(k,k)))/2.0
      if (h.lt.0.0) goto 121
      mx(k)=h
      goto 120
 121  mx(k)=h-1
 120  continue
c     scale, invert and then scale back
      is=0
 123  is=is+1
      do 122 k=1,nb
      do 122 j=1,nb
      xpx(k,j)=xpx(k,j)*10.0**(mx(k)+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 k=1,nb
      ob(k)=0.0
      do 133 j=1,nb
      ob(k)=ob(k)+xpx(k,j)*xpy(j)
 133  continue
      ss=0.0
      do 134 i=1,nn
      do 134 l=1,nt
      if (xx(i,l,1).ne.0.0) then
      ee=yy(i,l)
      do 135 k=1,nb
      ee=ee-xx(i,l,k)*ob(k)
 135  continue
      ss=ss+ee**2
      endif
 134  continue
      ob(nb+1)=ss/dfloat(nob-nb)
      do 136 k=1,nb
```

Appendix 6: Asymptotic Standard Errors of the COLS Estimators

The derivation of the standard errors of the COLS estimators follows the method used in the Appendix of Olsen, Schmidt and Waldman (1980). Consider the composed error disturbance term ε = V-U, where V~N(0, σ_V^2) and U~|N(0,σ^2)|. Aigner, Lovel and Schmidt (1977) show that the expectation of ε is

$$E(\varepsilon) = \mu_1 = -\sigma\sqrt{2/\pi} \tag{A6.1}$$

and Olsen, Schmidt and Waldman (1980) present the first six central moments of ε in terms of σ_V^2 and σ^2 . Their results can also be expressed in terms of σ_S^2 and γ by noting that σ_V^2 = (1-γ) σ_S^2 and σ^2 = $\gamma\sigma_S^2$. Substitution and rearrangement of terms yields

$$\mu_1 = -\sqrt{2\gamma\sigma_S^2/\pi} \tag{A6.2}$$

$$\mu_2 = \sigma_S^2\left[(1-\gamma)+\gamma\frac{(\pi-2)}{\pi}\right] \tag{A6.3}$$

$$\mu_3 = \sigma_S^3\left[\sqrt{\frac{2}{\pi}}\left(1-\frac{4}{\pi}\right)\gamma^{3/2}\right] \tag{A6.4}$$

$$\mu_4 = \sigma_S^4\left[3(1-\gamma)^2+6\frac{(\pi-2)}{\pi}\gamma(1-\gamma)+\left(3-\frac{4}{\pi}-\frac{12}{\pi^2}\right)\gamma^2\right] \tag{A6.5}$$

$$\mu_5 = \sigma_S^5\gamma^{3/2}\sqrt{\frac{2}{\pi}}\left[10(1-\frac{4}{\pi})(1-\gamma)+\left(7-\frac{20}{\pi}-\frac{16}{\pi^2}\right)\gamma\right] \tag{A6.6}$$

$$\mu_6 = \sigma_S^6\left[15(1-\gamma)^3+45\frac{(\pi-2)}{\pi}(1-\gamma)^2\gamma+15\left(3-\frac{4}{\pi}-\frac{12}{\pi^2}\right)(1-\gamma)\gamma^2\right.$$

$$\left.+\left(15-\frac{6}{\pi}-\frac{100}{\pi^2}-\frac{40}{\pi^3}\right)\gamma^3\right] \tag{A6.7}$$

These expressions will be used in the following asymptotic variances and covariances of the second and third sample central moments:

$$V(m_2) = \frac{1}{N}\left(\mu_4 - \mu_2^2\right) \tag{A6.8}$$

$$V(m_3) = \frac{1}{N}\left(\mu_6 - \mu_3^2 - 6\mu_2\mu_4 + 9\mu_2^3\right) \tag{A6.9}$$

$$Cov(m_2,m_3) = \frac{1}{N}\left(\mu_5 - 4\mu_2\mu_3\right). \tag{A6.10}$$

Before these expressions can be of use, we must re-express the COLS estimators of γ and β_0 in equations (8.4) and (8.5) (in Chapter 8) to be functions of the second and third moments. These become:

$$\hat{\gamma} = \left\{ m_2 \left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi - 4)} m_3 \right]^{-2/3} + \frac{2}{\pi} \right\}^{-1} \tag{A6.11}$$

$$\hat{\beta}_0 = \hat{\beta}_0(OLS) + \sqrt{\frac{2}{\pi}} \left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi - 4)} m_3 \right]^{1/3} \tag{A6.12}$$

and the estimator of σ_S^2 from equation (8.3) remains:

$$\hat{\sigma}_S^2 = m_2 + \frac{2}{\pi} \left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi - 4)} m_3 \right]^{2/3}. \tag{A6.13}$$

We now use Taylor-series expansions, truncated after the first-order terms, to obtain approximations of the asymptotic standard errors of the COLS estimators. To do this for $\hat{\beta}_0$ we need to specify the asymptotic covariance between $\hat{\beta}_0$ (OLS) and the third sample moment. Olsen, Schmidt and Waldman (1980) note that it is equal to zero when the model contains an intercept and no other regressors. Assuming this result extends to the general case, we obtain:

$$V(\hat{\beta}_0) = \left[\frac{\partial \hat{\beta}_0}{\hat{\beta}_0(OLS)} \right]^2 \cdot V\left(\hat{\beta}_0(OLS)\right) + \left[\frac{\partial \hat{\beta}_0}{\partial m_3} \right]^2 \cdot V(m_3) \tag{A6.14}$$

$$V\left(\hat{\sigma}_S^2\right) = \left[\frac{\partial \hat{\sigma}_S^2}{\partial m_2} \right]^2 \cdot V(m_2) + \left[\frac{\partial \hat{\sigma}_S^2}{\partial m_3} \right]^2 \cdot V(m_3) + \left[\frac{\partial \hat{\sigma}_S^2}{\partial m_2} \right] \left[\frac{\partial \hat{\sigma}_S^2}{\partial m_3} \right] Cov(m_2, m_3) \tag{A6.15}$$

$$V(\hat{\gamma}) = \left[\frac{\partial\hat{\gamma}}{\partial m_2}\right]^2 \cdot V(m_2) + \left[\frac{\partial\hat{\gamma}}{\partial m_3}\right]^2 \cdot V(m_3) + \left[\frac{\partial\hat{\gamma}}{\partial m_2}\right]\left[\frac{\partial\hat{\gamma}}{\partial m_3}\right] Cov(m_2, m_3), \qquad \text{(A6.16)}$$

where

$$\left[\frac{\partial\hat{\beta}_0}{\partial\hat{\beta}_0(OLS)}\right] = 1 \qquad \text{(A6.17)}$$

$$\left[\frac{\partial\hat{\beta}_0}{\partial m_3}\right] = \frac{\pi}{(\pi-4)} \cdot \frac{1}{3}\left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} m_3\right]^{-2/3} \qquad \text{(A6.18)}$$

$$\left[\frac{\partial\hat{\sigma}_S^2}{\partial m_2}\right] = 1 \qquad \text{(A6.19)}$$

$$\left[\frac{\partial\hat{\sigma}_S^2}{\partial m_3}\right] = \sqrt{\frac{2}{\pi}} \cdot \frac{\pi}{(\pi-4)} \cdot \frac{2}{3}\left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} m_3\right]^{-1/3} \qquad \text{(A6.20)}$$

$$\left[\frac{\partial\hat{\gamma}}{\partial m_2}\right] = -\left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} m_3\right]^{-2/3} \left\{m_3\left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} m_3\right]^{-2/3} + \frac{2}{\pi}\right\}^{-2} \qquad \text{(A6.21)}$$

$$\left[\frac{\partial\hat{\gamma}}{\partial m_3}\right] = m_2\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} \cdot \frac{2}{3}\left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} m_3\right]^{-5/3} \left\{m_3\left[\sqrt{\frac{\pi}{2}} \cdot \frac{\pi}{(\pi-4)} m_3\right]^{-2/3} + \frac{2}{\pi}\right\}^{-2}$$

(A6.22)

Appendix 7: SHAZAM Code For COLS

This code assumes a data file (eg.dta) which contains three columns of data, each of length 60, on output and two inputs. The code obtains OLS estimates of a Cobb-Douglas production function. It then calculates the third-moment test statistic before calculating the COLS estimates and their standard errors.

```
sample 1 60
read(eg.dta) y x1 x2
genr ly=log(y)
genr lx1=log(x1)
genr lx2=log(x2)
ols ly lx1 lx2/resid=e coef=b stderr=se
genl b0ols=b:3
genl vb0ols=(se:3)**2
genr e2=e**2
stat e2/mean=m2
genr e3=e**3
stat e3/mean=m3
genl n=60
genl m3t=m3/sqrt(6*m2**3/n)
* third moment test:
print m3t
genl pi=3.142
genl s2=m2+2/pi*(sqrt(pi/2)*pi/(pi-4)*m3)**(2/3)
genl g=(sqrt(pi/2)*pi/(pi-4)*m3)**(2/3)/s2
genl b0=b0ols+sqrt(2*g*s2/pi)
* COLS estimates:
print s2 g b0
genl u2=s2*((1-g)+g*(pi-2)/pi)
genl u3=s2**(3/2)*(sqrt(2/pi)*(1-4/pi)*g**(3/2))
genl u4=s2**2*(3*(1-g)**2+6*(pi-2)/pi*g*(1-g)+(3-4/pi-12/pi**2)*g**2)
genl u5=s2**(5/2)*g**(3/2)*sqrt(2/pi)*(10*(1-4/pi)*(1-g)+(7-20/pi-16/pi**2)*g)
genl u6=s2**3*(15*(1-g)**3+45*(pi-2)/pi*(1-g)**2*g &
+15*(3-4/pi-12/pi**2)*(1-g)*g**2+(15-6/pi-100/pi**2-40/pi**3)*g**3)
genl vm2=1/n*(u4-u2**2)
genl vm3=1/n*(u6-u3**2-6*u2*u4+9*u2**3)
genl cov=1/n*(u5-4*u2*u3)
genl tmp=sqrt(pi/2)*pi/(pi-4)*m3
genl db0dm3=pi/(pi-4)/3*tmp**(-2/3)
genl ds2dm3=sqrt(2/pi)*pi/(pi-4)*2/3*tmp**(-1/3)
genl dgdm2=-(tmp**(-2/3))*(tmp**(-2/3)+2/pi)**(-2)
genl dgdm3=m2*sqrt(pi/2)*pi/(pi-4)*2/3*tmp**(-5/3)*(m2*tmp**(-2/3)+2/pi)**(-2)
genl seb0=sqrt(vb0ols+db0dm3**2*vm3)
genl ses2=sqrt(vm2+ds2dm3**2*vm3+2*ds2dm3*cov)
genl seg=sqrt(dgdm2**2*vm2+dgdm3**2*vm3+2*dgdm2*dgdm3*cov)
* standard errors of COLS estimates:
print seb0 ses2 seg
stop
```

References

Afriat, S.N. (1972), "Efficiency Estimation of Production Functions", *International Economic Review*, 13, 568-598.

Aigner, D.J. and S.F. Chu (1968), "On Estimating the Industry Production Function", *American Economic Review*, 58, 826-839.

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

Ali, A.I. and L.M. Seiford (1993), "The Mathematical Programming Approach to Efficiency Analysis", in Fried, H.O., C.A.K. Lovell and S.S. Schmidt (Eds), *The Measurement of Productive Efficiency: Techniques and Applications*, Oxford University Press, New York, pp. 120-159.

Ali, M. and J.C. Flinn (1989), "Profit Efficiency Among Basmati Rice Producers in Pakistan Punjab", *American Journal of Agricultural Economics*, 71, 303-310.

Ali, M. and M.A. Chaudhry (1990), "Inter-regional Farm Efficiency in Pakistan's Punjab: A Frontier Production Function Study", *Journal of Agricultural Economics*, 41, 62-74.

Aly, H.Y., K. Belbase, R. Grabowski and S. Kraft (1987), "The Technical Efficiency of Illinios Grain Farms: An Application of a Ray-Homothetic Production Function", *Southern Journal of Agricultural Economics*, 19, 69-78.

Atkinson, S.E. and Halvorsen, R. (1984), "Parametric Efficiency Tests, Economies of Scale, and Input Demand in U.S. Electric Power Generation", *International Economic Review*, 25, 647-662.

Bailey, D.V., B. Biswas, S.C. Kumbhakar and B.K. Schulthies (1989), "An Analysis of Technical, Allocative, and Scale Inefficiency: The Case of Ecuadorian Dairy Farms", *Western Journal of Agricultural Economics*, 14, 30-37.

Baltagi, B., Y. Chang and Q. Li (1992), "Monte Carlo Results on Several New and Existing Tests for the Error Components Model", *Journal of Econometrics*, 54, 95-120.

Banker, R.D., A. Charnes and W.W. Cooper (1984), "Some Models for Estimating Technical and Scale Inefficiency in Data Envelopment Analysis", *Management Science*, 30, 1078-1092.

Banker, R.D., A. Charnes, W.W. Cooper and A. Maindiratta (1988), "A Comparison of DEA and Translog Estimates of Production Functions using Simulated Data from a Known Technology" in A. Dogramaci and R. Färe (eds), *Applications of Modern Production Theory: Efficiency and Productivity*, Kluwer, Boston.

Banker, R.D. and R.C. Morey (1986a), "Efficiency Analysis for Exogenously Fixed Inputs and Outputs", *Operations Research*, 34, 513-521.

Banker, R.D. and R.C. Morey (1986b), "The Use of Categorical Variables in Data Envelopment Analysis", *Management Science*, 32, 1613-1627.

Bardhan, P.K. (1973), "Size, Productivity and Returns to Scale: An Application of Farm-level Data in Indian Agriculture", *Journal of Political Economy*, 81, 1370-1386.

Bateson, J. and P.L. Swan (1989), "Economies of Scale and Utilisation: An analysis of the Multi-Plant Generation Costs of the Electricity Commission of New South Wales, 1970/71-1984/85", *Economic Record*, 65, 329-344.

Battese, G.E. (1992), "Frontier Production Functions and Technical Efficiency: A Survey of Empirical Applications in Agricultural Economics", *Agricultural Economics*, 7, 185-208.

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

Battese, G.E. and T.J. Coelli (1992), "Frontier Production Functions, Technical Efficiency and Panel Data: With Application to Paddy Farmers in India", *Journal of Productivity Analysis*, 3, 153-169.

Battese, G.E. and T.J. Coelli (1993), "A Stochastic Frontier Production Function Incorporating a Model for Technical Inefficiency Effects", *Working Papers in Econometrics and Applied Statistics*, No. 69, Department of Econometrics, University of New England, Armidale, pp. 22.

Battese, G.E. and T.J. Coelli (1995), "A Model for Technical Inefficiency Effects in a Stochastic Frontier Production Function for Panel Data", *Empirical Economics*, 20, 325-332.

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.

Battese, G.E., S.J. Malik and S. Broca (1993), "Production Functions for Wheat Farmers in Selected Districts of Pakistan: An Application of a Stochastic Frontier Production Function with Time-varying Inefficiency Effects", *Pakistan Development Review*, 32, 233-268.

Battese, G.E., S.J. Malik and M.A. Gill (1996), "An Investigation of Technical Inefficiencies of Production of Wheat Farmers in Four Districts of Pakistan", *Journal of Agricultural Economics*, 47, 37-49.

Battese, G.E. and G.A. Tessema (1993), "Estimation of Stochastic Frontier Production Functions with Time-varying Parameters and Technical Efficiencies using Panel Data from Indian Villages", *Agricultural Economics*, 9, 313-333.

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.

Belbase, K. and R. Grabowski (1985), "Technical Efficiency in Nepalese Agriculture", *Journal of Developing Areas*, 19, 515-525.

Berndt, E.R. (1978), "Aggregate Energy, Efficiency, and Productivity Measurement", *Annual Review of Energy*, 3, 225-273.

Bernstein, M.A., S.L. Feldman and A.P. Schinnar (1990), "Impact of Pollution Controls on the Productivity of Coal-Fired Power Plants", *Energy Economics*, 14, 11-17.

Binswanger, H.P. and N.S. Jodha (1978), *Manual of Instructions for Economic Investigators in ICRISAT's Village Level Studies, Volume II*, Village Level Studies Series, Economics Program, International Crops Research Institute for the Semi-Arid Tropics, Patancheru, Andhra Pradesh, India.

Bravo-Ureta, B.E. (1986), "Technical Efficiency Measures for Dairy Farms Based on a Probabilistic Frontier Function Model", *Canadian Journal of Agricultural Economics*, 34, 399-415.

Bravo-Ureta, B.E. and R.E. Evenson (1994), "Efficiency in Agricultural Production: The Case of Peasant Farmers in Eastern Paraguay", *Agricultural Economics*, 10, 27-37.

Bravo-Ureta, B.E. and A.E. Pinheiro (1993), "Efficiency Analysis of Developing Country Agriculture: A Review of the Frontier Function Literature", *Agricultural and Resource Economics Review*, 22, 88-101.

Bravo-Ureta, B.E. and L. Rieger (1990), "Alternative Production Frontier Methodologies and Dairy Farm Efficiencies", *Journal of Agricultural Economics*, 41, 215-226.

Bravo-Ureta, B.E. and L. Rieger (1991), "Dairy Farm Efficiency Measurement Using Stochastic Frontiers and Neoclassical Duality", *American Journal of Agricultural Economics*, 73, 421-428.

Charnes, A., W.W. Cooper and E. Rhodes (1978), "Measuring the Efficiency of Decision Making Units", *European Journal of Operations Research*, 2, 429-444.

Charnes, A., W.W. Cooper, J. Rousseau and J. Semple (1987), "Data Envelopment Analysis and Axiomatic Notions of Efficiency and Reference Sets", *Research Report CCS 558*, Centre for Cybernetic Studies, The University of Texas at Austin.

Charnes, A., W.W. Cooper, L. Sieford and J. Stutz (1982), "A Multiplicative Model for Efficiency Analysis", *Socio-Economic Planning Sciences*, 16, 223-224.

Charnes, A., W.W. Cooper, L. Sieford and J. Stutz (1983), "Invariant Multiplicative Efficiency and Piecewise Cobb-Douglas Envelopments", *Operations Research Letters*, 2, 101-103.

Chavas, J. and M. Aliber (1993), "An Analysis of Economic Efficiency in Agriculture: A Nonparametric Approach", *Journal of Agricultural and Resource Economics*, 18, 1-16,

Christensen, L.R. and W.H. Greene (1976), "Economies of Scale in U.S. Electric Power Generation", *Journal of Political Economy*, 84, 655-676.

Coelli, T.J. (1992), "A Computer Program for Frontier Production Function Estimation: FRONTIER, Version 2.0", *Economics Letters*, 39, 29-32.

Coelli, T.J. (1993), "Finite Sample Properties of Stochastic Frontier Estimators and Associated Test Statistics", *Working Papers in Econometrics and Applied Statistics,* No. 70, Department of Econometrics, University of New England, Armidale, pp. 30.

Coelli, T.J. (1994), *A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation*, mimeo, Department of Econometrics, University of New England, Armidale, pp. 32.

Coelli, T.J. (1995a), "Recent Developments in Frontier Estimation and Efficiency Measurement", *Australian Journal of Agricultural Economics*, (to appear).

Coelli, T.J. (1995b), *The Measurement and Sources of Productive Efficiency in Electricity Generation in Australia*, Report to the AESIRB, Department of Econometrics, University of New England, Armidale, pp. 36.

Coelli, T.J. (1995c), "A Monte Carlo Analysis of the Stochastic Frontier Production Function", *Journal of Productivity Analysis*, 6, 247-268.

Coelli, T.J. and G.E. Battese (1994), "Identification of Factors Which Influence the Technical Inefficiency of Indian Farmers", *Working Papers in Econometrics and Applied Statistics*, No. 76, Department of Econometrics, University of New England, Armidale, pp. 24.

Cornwell, C., P. Schmidt and R.C. Sickles (1990), "Production Frontiers With Cross-sectional and Time-series Variation in Efficiency Levels", *Journal of Econometrics*, 46, 185-200.

Courville, L. (1974), "Regulation and Efficiency in the Electric Utility Industry", *Bell Journal of Economics and Management Science*, 5, 53-74.

Cowing, T.G. and V.K. Smith (1978), "The Estimation of a Production Technology: A Survey of Econometric Analyses of Steam-Electric Generation", *Land Economics*, 54, 156-186.

D'Agostino, R.B. and E.S. Pearson (1973), "Tests for Departure from Normality: Empirical Results for the Distribution of b_2 and $\sqrt{b_1}$", *Biometrika*, 60, 613-622.

Dawson, P.J. (1985), "Measuring Technical Efficiency from Production Functions: Some Further Estimates", *Journal of Agricultural Economics*, 36, 31-40.

Dawson, P.J. and J. Lingard (1989), "Measuring Farm Efficiency over Time on Philippine Rice Farms", *Journal of Agricultural Economics*, 40, 168-177.

Dawson, P.J., J. Lingard and C.H. Woodford (1991), "A Generalised Measure of Farm-Specific Technical Efficiency", *American Journal of Agricultural Economics*, 73, 1098-1104.

Debreu, G. (1951), "The Coefficient of Resource Utilisation", *Econometrica*, 19, 273-292.

Defourny, J., C.A.K. Lovell and A.G.M. N'gbo (1991), "Variation in Productive Efficiency in French Workers' Cooperatives", *Journal of Productivity Analysis*", 3, 103-117.

Deolalikar, A.B. and W.P.M. Vijverberg (1983), "The Heterogeneity of Family and Hired Labor in Agricultural Production: A Test Using District-Level Data from India", *Journal of Economic Development*, 8, 45-69.

Ekanayake, S.A.B. (1987), "Location Specificity, Settler Type and Productive Efficiency: A Study of the Mahaweli Project in Sri Lanka", *Journal of Development Studies*, 23, 509-521.

Ekanayake, S.A.B. and S.K. Jayasuriya (1987), "Measurement of Farm-Specific Technical Efficiency: A Comparison of Methods", *Journal of Agricultural Economics*, 38, 115-122.

Färe, R., R. Grabowski and S. Grosskopf (1985), "Technical Efficiency of Philippine Agriculture", *Applied Economics*, 17, 205-214.

Färe, R., S. Grosskopf and J. Logan (1983), "The Relative Efficiency of Illinios Electric Utilities", *Resources and Energy*, 5, 349-367.

Färe, R., S. Grosskopf and J. Logan (1984), "The Relative Performance of Publicly-Owned and Privately-Owned Electric Utilities", *Journal of Public Economics*, 26, 89-106.

Färe, R., S. Grosskopf and C.A.K. Lovell (1985), *The Measurement of Efficiency of Production*, Kluwer-Nijhoff, Boston.

Färe, R., S. Grosskopf, M. Norris and Z. Zhang (1994), "Productivity Growth, Technical Progress, and Efficiency Changes in Industrialised Countries", *American Economic Review*, 84, 66-83.

Färe, R., S. Grosskopf and C. Pasurka (1986), "Effects on Relative Efficiency in Electric Power Generation Due to Environmental Controls", *Resources and Energy*, 8, 167-184.

Färe, R., S. Grosskopf and C. Pasurka (1989), "The Effect of Environmental Regulations on the Efficiency of Electric Utilities: 1969 versus 1975", *Applied Economics*, 21, 225-235.

Färe, R. and C.A.K. Lovell (1978), "Measuring the Technical Efficiency of Production", *Journal of Economic Theory*, 19, 150-162.

Farrell, M.J. (1957), "The Measurement of Productive Efficiency", *Journal of the Royal Statistical Society*, Series A, 120, 253-290.

Ferrier, G.D. and C.A.K. Lovell (1990), "Measuring Cost Efficiency in Banking: Econometric and Linear Programming Evidence", *Journal of Econometrics*, 46, 229-245.

Førsund, F.R. and L. Hjalmarsson (1979), "Generalised Farrell Measures of Efficiency: An Application to Milk Processing in Swedish Dairy Plants", *Economic Journal*, 89, 294-315.

Førsund, 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.

Gimbol, K.C., G.E. Battese and E.M. Fleming (1996), "Technical efficiencies of Smallholder Cocoa Producers in Papua New Guinea: A Stochastic Frontier Analysis", *Quarterly Journal of International Agriculture*, (to appear).

Gollop, F.M. and M.J. Roberts (1983), "Environmental Regulations and Productivity Growth: The Case of Fossil-Fuelled Electric Power Generation", *Journal of Political Economy*, 91, 654-674.

Gong, B. and R.C. Sickles (1989), "Finite Sample Evidence on the Performance of Stochastic Frontier Models using Panel Data", *Journal of Productivity Analysis*, 1, 229-261.

Gong, B. and R.C. Sickles (1992), "Finite Sample Evidence on the Performance of Stochastic Frontiers and Data Envelopment Analysis using Panel Data", *Journal of Econometrics*, 51, 259-284.

Gourieroux, C., A. Holly and A. Monfort (1982), "Likelihood Ratio Test, Wald Test, and Kuhn-Tucker Test in Linear Models with Inequality Constraints on the Regression Parameters", *Econometrica*, 50, 63-80.

Greene, W.H. (1980a), "Maximum Likelihood Estimation of Econometric Frontier Functions", *Journal of Econometrics*, 13, 27-56.

Greene, W.H. (1980b), "On the Estimation of a Flexible Frontier Production Model", *Journal of Econometrics*, 13, 101-116.

Greene, W.H. (1990), "A Gamma-distributed Stochastic Frontier Model", *Journal of Econometrics*, 46, 141-164.

Greene, W.H. (1992), *LIMDEP Version 6.0: User's Manual and Reference Guide*, Econometric Software Inc., New York.

Greene, W.H. (1993), "The Econometric Approach to Efficiency Analysis", in Fried, H.O., C.A.K. Lovell and S.S. Schmidt (Eds), *The Measurement of Productive Efficiency: Techniques and Applications*, Oxford University Press, New York, pp. 68-119.

Hammond, C.J. (1992), "Privatisation and the Efficiency of Decentralised Electricity Generation: Some Evidence from Inter-War Britain", *The Economic Journal*, 102, 538-553.

Hausman, W.J. and J.L. Neufeld (1991), "Property Rights Versus Public Spirit: Ownership and Efficiency of U.S. Electric Utilities Prior to Rate-of-Return Regulation", *Review of Economics and Statistics*, 57, 414-423.

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

Honda, Y. (1985), "Testing the Error Components Model with Non-Normal Disturbances", *Review of Economic Studies*, 52, 681-690.

Huang, C. and J-T. Liu (1994), "Estimation of a Non-neutral Stochastic Frontier Production Function", *Journal of Productivity Analysis*, 4, 171-180.

Huang, C.J., A.M. Tang and F.S. Bagi (1986), "Two Views of Efficiency in Indian Agriculture", *Canadian Journal of Agricultural Economics*, 34, 209-226.

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

Jensen, U. (1993), "Measuring Individual Inefficiency with Frontier Functions: A Monte Carlo Study", *Discussion Paper 68*, Institut fur Statistic und Okonometrie, der Universitat Kiel, Kiel, pp. 18.

Jondrow, J., C.A.K. Lovell, I.S. Materov and P. Schmidt (1982), "On Estimation of Technical Inefficiency in the Stochastic Frontier Production Function Model", *Journal of Econometrics*, 19, 233-238.

Judge, G.G., W.E. Griffiths, R. C. Hill, H. Lutkepohl and T.C. Lee (1985), *The Theory and Practice of Econometrics*, Second Edition, Wiley, New York.

Kalirajan, K.P. (1981), "An Econometric Analysis of Yield Variability in Paddy Production", *Canadian Journal of Agricultural Economics*, 29, 283-294.

Kalirajan, K.P. (1982), "On Measuring Yield Potential of the High Yielding Varieties Technology at Farm Level", *Journal of Agricultural Economics* 33, 227-236.

Kalirajan, K.P. (1989), "On Measuring the Contribution of Human Capital to Agricultural Production", *Indian Economic Review*, 24, 247-261.

Kalirajan, K.P. (1991), "The Importance of Efficient Use in the Adoption of Technology: A Micro Panel Data Analysis", *Journal of Productivity Analysis*, 2, 113-126.

Kalirajan, K.P. and J.C. Flinn (1983), "The Measurement of Farm-specific Technical Efficiency", *Pakistan Journal of Applied Economics,* 2, 167-180.

Kalirajan, K.P. and R.T. Shand (1986), "Estimating Location-Specific and Firm-Specific Technical Efficiency: An Analysis of Malaysian Agriculture", *Journal of Economic Development*, 11, 147-160.

Kalirajan, K.P. and R.T. Shand (1989), "A Generalised Measure of Technical Efficiency", *Applied Economics*, 21, 25-34.

Kodde, D.A. and F.C. Palm (1986), "Wald Criteria for Jointly Testing Equality and Inequality Restrictions", *Econometrica*, 54, 1243-1248.

Komiya, R. (1962), "Technical Progress and the Production Function in the United States Steam Power Industry", *Review of Economics and Statistics*, 44, 156-166.

Koopmans, T.C. (1951), "An Analysis of Production as an Efficient Combination of Activities", in T.C. Koopmans, Ed., *Activity Analysis of Production and Allocation*, Cowles Commission for Research in Economics, Monograph No. 13, Wiley, New York.

Kopp, R.J. and W.E. Diewert (1982), "The Decomposition of Frontier Cost Function Deviations into Measures of Technical and Allocative Efficiency", *Journal of Econometrics*, 19, 319-331.

Kopp, R.J. and Smith V.K. (1980), "Frontier Production Function Estimates for Steam Electric Generation: A Comparative Analysis", *Southern Economic Journal*, 46, 1049-1059.

Kumbhakar, S.C. (1990), "Production Frontiers, Panel Data and Time-Varying Technical Inefficiency", *Journal of Econometrics*, 46, 201-211.

Kumbhakar, S.C. (1994), "Efficiency Estimation in a Profit Maximising Model Using Flexible Production Function", *Agricultural Economics*, 10, 143-152.

Kumbhakar, S.C. (1996), "Modeling Allocative Inefficiency in a Translog Cost Function and Cost Share Equations: An Exact Relationship", *Journal of Econometrics*, (to appear).

Kumbhakar, S.C., B. Biswas and D.V. Bailey (1989), "A Study of Economic Efficiency of Utah Dairy Farmers: A System Approach", *The Review of Economics and Statistics*, 71, 595-604.

Kumbhakar, S.C., S.C. Ghosh and J.T. McGuckin (1991), "A Generalised Production Frontier Approach for Estimating Determinants of Inefficiency in U.S. Dairy Farms", *Journal of Business and Economic Statistics*, 9, 279-286.

Lawrence, D., Swan, P.L. and Zeitsch, J. (1990), "The Comparative Efficiency of State Electricity Authorities", *Working Papers Series*, No. 90-034, Australian Graduate School of Management, University of New South Wales, Sydney.

Lee, L-F. (1993), "Asymptotic Distribution for the Maximum Likelihood Estimator for a Stochastic Frontier Function Model with a Singular Information Matrix", *Econometric Theory*, 9, 413-430.

Lee, Y.H. and P. Schmidt (1993), "A Production Frontier Model with Flexible Temporal Variation in Technical Inefficiency", in Fried, H.O., C.A.K. Lovell and S.S. Schmidt (Eds), *The Measurement of Productive Efficiency: Techniques and Applications*, Oxford University Press, New York, pp. 237-255.

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

Lovell, C.A.K. (1993), "Production Frontiers and Productive Efficiency", in Fried, H.O., C.A.K. Lovell and S.S. Schmidt (Eds), *The Measurement of Productive Efficiency Techniques and Applications*, Oxford University Press, New York, 3-67.

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

Nemoto, J. Y. Nakanishi and S. Madono (1993), "Scale Economies and Over-Capitalisation in Japanese Electric Utilities", *International Economic Review*, 34, 431-440.

Nerlove, M. (1963), "Returns to Scale in Electricity Supply", in Christ, C.F. (Ed), *Measurement in Economics, Studies in Mathematical Economics and Econometrics in Honor of Yehuda Grunfeld*, Stanford University Press.

Olsen, J.A., P. Schmidt and D.M. Waldman (1980), "A Monte Carlo Study of Estimators of the Stochastic Frontier Production Function", *Journal of Econometrics*, 13, 67-82.

Pagan, A.R. and A.D. Hall (1983), "Diagnostic Tests as Residual Analysis", *Econometric Reviews*, 2, 159-218.

Parikh, A. and K. Shah (1994), "Measurement of Technical Efficiency in the North West Frontier Province of Pakistan", *Journal of Agricultural Economics*, 45, 132-138.

Phillips, M.J. and R.P. Marble (1986), "Farmer Education and Efficiency: A Frontier Production Function Approach", *Economics of Education Review*, 5, 257-264.

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.

Price, D.E., N. Jip, K.M. Woods, C. Farah, R.F. O'Brien and J.E. Pierce (1992), "A Comparison of Power Station Productivity in New South Wales", *Conference Paper*, no. 3, Pacific Power, Sydney.

Rawlins, G. (1985), "Measuring the Impact of I.R.D.P.II upon the Technical Efficiency Level of Jamaican Peasant Farmers", *Social and Economic Studies*, 34, 71-96.

Ray, S (1985), " Measurement and Test of Efficiency of Farms in Linear Programming Models: A Study of West Bengal Farms", *Oxford Bulletin of Economics and Statistics*, 47, 371-386.

Reifschneider, D. and R. Stevenson (1991), "Systematic Departures from the Frontier: A Framework for the Analysis of Firm Inefficiency", *International Economic Review*, 32, 715-723.

Richmond, J. (1974), "Estimating the Efficiency of Production", *International Economic Review*, 15, 515-521.

Saini, G.R. (1979), *Farm Size, Resource-Use Efficiency and Income Distribution*, Allied Publishers, New Delhi.

Schmidt, P. (1976), "On the Statistical Estimation of Parametric Frontier Production Functions", *Review of Economics and Statistics*, 58, 238-239.

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

Schmidt, P. and C.A.K. Lovell (1979), "Estimating Technical and Allocative Inefficiency Relative to Stochastic Production and Cost Functions", *Journal of Econometrics*, 9, 343-366.

Schmidt, P. and C.A.K. Lovell (1980), "Estimating Stochastic Production and Cost Frontiers when Technical and Allocative Inefficiency are Correlated", *Journal of Econometrics*, 13, 83-100.

Schmidt, P. and T. Lin (1984), "Simple Tests of Alternative Specifications in Stochastic Frontier Models", *Journal of Econometrics*, 24, 349-361.

Schmidt, P. and R.C. Sickles (1984), "Production Frontiers and Panel Data", *Journal of Business and Economic Statistics*, 2, 367-374.

Seiford, L.M. and R.M. Thrall (1990), "Recent Developments in DEA: The Mathematical Approach to Frontier Analysis", *Journal of Econometrics*, 46, 7-38.

Seitz, W.D. (1971), "Productive Efficiency in the Steam-Electric Generating Industry", *Journal of Political Economy*, 79, 879-886.

Sengupta, J.K. (1990), "Transformations in Stochastic DEA Models", *Journal of Econometrics*, 46, 109-124.

Squires, D. and S. Taber (1991), "Technical Efficiency and Future Production Gains in Indonesian Agriculture", *The Developing Economies*, 29, 258-270.

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

Stewart, J.F. (1979), "Plant Size, Plant Factor, and the Shape of the Average Cost Function in Electric Power Generation: A Nonhomogenous Capital Approach", *The Bell Journal of Economics*, 10, 332-344.

Swan, P.L. (1990), "Real Rates of Return in Electricity Supply: New South Wales, Tasmania and Victoria", *Economic Record*, 66, 93-109.

Tauer, L.W. and K.P. Belbase (1987), "Technical efficiency of New York Dairy Farms", *Northeastern Journal of Agricultural and Resource Economics*, 16, 10-16.

Taylor, T.G., H.E. Drummond and A.T. Gomes (1986), "Agricultural Credit Programs and Production Efficiency: An Analysis of Traditional Farming in Southeastern Minas Gerais, Brazil", *American Journal of Agricultural Economics*, 68, 110-119.

Taylor, T.G. and J.S. Shonkwiler (1986), "Alternative Stochastic Specifications of the Frontier Production Function in the Analysis of Agricultural Credit Programs and Technical Efficiency", *Journal of Development Economics*, 21, 149-160.

Timmer, C.P. (1971), "Using a Probabilistic Frontier Function to Measure Technical Efficiency", *Journal of Political Economy*, 79, 776-794.

Tobin, J. (1958), "Estimation of Relationships for Limited Dependent Variables", *Econometrica*, 26, 24-36.

Tran, S., T. Coelli and E. Fleming (1993), "Analysis of the Technical Efficiency of State Rubber Farms in Vietnam", *Agricultural Economics*, 9, 183-201.

Waldman, D. (1982), "A Stationary Point for the Stochastic Frontier Likelihood", Journal of Econometrics, 18, 275-280.

Walker, T.S. and J.G. Ryan (1990), *Village and Household Economies in India's Semi-arid Tropics*, The John Hopkins University Press, Baltimore.

White, K. (1993), *SHAZAM User's Reference Manual Version 7.0*, McGraw-Hill.

Zellner, A., J. Kmenta and J. Drèze (1966), "Specification and Estimation of Cobb-Douglas Production Function Models", *Econometrica*, 34, 784-795.

Zellner, A. and N.S. Revankar (1969), "Generalized Production Functions", *Review of Economic Studies*, 36, 241-250.

Zieschang, K.D. (1983), "A Note on the Decomposition of Cost Efficiency into Technical and Allocative Components", *Journal of Econometrics*, 23, 401-405.