Estimating the CES Function in R: Package micEconCES

Géraldine Henningsen
Danmarks Statistik

Arne Henningsen
University of Copenhagen

Abstract

The Constant-Elasticity-of-Substitution (CES) function is popular in several areas of economics but it is rarely used in econometric analysis, because it cannot be estimated by standard linear regression techniques. We discuss several approaches to estimate the CES function and demonstrate how they can be applied in R using the add-on package **micEconCES**. Furthermore, we describe how we implemented the various estimation approaches in the **micEconCES** package and we compare them in a Monte Carlo study. Given the data generating process used in our analysis, all estimation approaches provided satisfying results.

Keywords: constant elasticity of substitution, CES, R.

1. Introduction

The Constant-Elasticity-of-Substitution (CES) function was developed as a generalisation of the Cobb-Douglas function by the Stanford group around Arrow, Chenery, Minhas, and Solow (1961). In recent years the CES has gained in importance in macroeconomics (e.g. Amras 2004; Bentolila and Gilles 2006) and growth theory (e.g. Caselli 2005; Caselli and Coleman 2006) as an alternative to the Cobb-Douglas function and it can be applied in many other fields. In microeconomics the CES function gained less popularity most likely because of its restrictive assumptions, especially in the case of more than two explanatory variables.

The formal specification of a CES production function[1] with two inputs is

$$y = \gamma \left(\delta x_1^{-\rho} + (1 - \delta) \, x_2^{-\rho} \right)^{-\frac{\nu}{\rho}}, \tag{1}$$

where y is the output quantity, x_1 and x_2 are the input quantities, and γ, δ, ρ, and ν are parameters. Parameter $\gamma \in (0, \infty)$ determines the productivity, $\delta \in (0, 1)$ determines the

[1] The CES functional form can be used to model different economic relationships (e.g. as production function or utility function). However, as the CES functional form is mostly used to model production technology, we name the independent (right-hand side) variables "inputs" and the dependent (left-hand side) variable "output" to keep the notation simple.

optimal distribution of the inputs, $\rho \in (-1, 0) \cup (0, \infty)$ determines the (constant) elasticity of substitution, which is $\sigma = 1/(1+\rho)$, and $\nu \in (0, \infty)$ is equal to the elasticity of scale.[2]

The CES function includes three special cases: for $\rho \to 0$, σ approaches 1 and the CES turns to the Cobb-Douglas form; for very large ρ, σ approaches 0 and the CES turns to the Leontief production function; and for $\rho \to -1$, σ approaches infinity and the CES turns to a linear function if ν is equal to 1.

As the CES function is non-linear in parameters and cannot be linearised analytically, it is not possible to estimate it with the usual linear estimation techniques. Therefore, the CES is usually approximated by the so-called "Kmenta approximation" (Kmenta 1967) or estimated by non-linear least-squares using different optimization algorithms. In this paper, we describe and compare these estimation approaches, explain how we implemented them in the R package **micEconCES** (Henningsen and Henningsen 2010), and show how they can be used for economic analysis and modelling. The **micEconCES** package is developed as part of the "micEcon" project on R-Forge (http://r-forge.r-project.org/projects/micecon/). Stable versions of this package are available for download from the Comprehensive R Archive Network (CRAN, http://CRAN.R-Project.org/package=micEconCES).

The paper is structured as follows. In the next section we discuss several approaches to estimate the CES production function and show how they can be applied in R. The third section describes the implementation of these methods in the R package **micEconCES**. Section four presents the results of a Monte Carlo study to compare the various estimation approaches, and the last section concludes.

2. Estimation of the CES production function

Tools for economic analysis with CES function are available in the R package **micEconCES** (Henningsen and Henningsen 2010). If this package is installed, it can be loaded with the command

```
> library("micEconCES")
```

We demonstrate the usage of this package by estimating a CES model with an artifical data set, because this avoids several problems that usually occur with real-world data.

```
> set.seed(123)
> cesData <- data.frame(x1 = rchisq(200, 10), x2 = rchisq(200,
+     10))
> cesData$y <- cesCalc(xNames = c("x1", "x2"), data = cesData,
```

[2] Originally, the CES function of Arrow *et al.* (1961) could model only constant returns to scale but later Kmenta (1967) added the parameter ν, which allows for variable returns to scale if $\nu \neq 1$.

```
+     coef = c(gamma = 1, delta = 0.6, rho = 0.5, nu = 1.1))
> cesData$y <- cesData$y + 2.5 * rnorm(200)
```

The first line sets the "seed" for the random number generator so that these examples can be replicated with exactly the same data set. The second line creates a data set with two input variables (called `x1` and `x2`) that have 200 observations each and are generated from a random χ^2 distribution with 10 degrees of freedom. The third line uses the command `cesCalc` that is included in the **micEconCES** package and calculates the deterministic output variable (called `y`) given the CES production function with the two input variables `x1` and `x2` and the coefficients $\gamma = 1$, $\delta = 0.6$, $\rho = 0.5$, and $\nu = 1.1$. The last line generates the stochastic output variable by adding normally distributed random errors to the deterministic output variable.

As the CES function is non-linear in its parameters, the most straightforward way to estimate the CES function in R would be to use `nls`, which performs non-linear least-squares estimations.

```
> cesNls <- nls(y ~ gamma * (delta * x1^(-rho) + (1 - delta) *
+     x2^(-rho))^(-phi/rho), data = cesData, start = c(gamma = 0.5,
+     delta = 0.5, rho = 0.25, phi = 1))
> print(cesNls)

Nonlinear regression model
  model:  y ~ gamma * (delta * x1^(-rho) + (1 - delta) * x2^(-rho))^(-phi/rho)
   data:  cesData
 gamma  delta    rho    phi
1.0102 0.6271 0.6398 1.0955
 residual sum-of-squares: 1175

Number of iterations to convergence: 6
Achieved convergence tolerance: 4.147e-07
```

While the `nls` routine works well in this ideal artificial example, it does not perform well in many applications with real data, either because of non-convergence, convergence to a local minimum, or theoretically unreasonable parameter estimates. Therefore, we show alternative ways of estimating the CES function in the following subsections.

2.1. Kmenta approximation

Given that non-linear estimation methods are often troublesome—particularly during the 1960s and 1970s when computing power was very limited—Kmenta (1967) derived an approximation of the classical two-input CES production function that can be estimated by

ordinary least-squares techniques.

$$\begin{aligned}\log y = &\log \gamma + \nu\, \delta \log x_1 + \nu\, (1-\delta) \log x_2 \\ &- \frac{\rho\, \nu}{2}\, \delta\, (1-\delta)\, (\log x_1 - \log x_2)^2\end{aligned} \tag{2}$$

While Kmenta (1967) obtained this formula by logarithmising the CES function and applying a second-order Taylor series expansion to $\log\left(\delta x_1^{-\rho} + (1-\delta) x_2^{-\rho}\right)$ at the point $\rho = 0$, the same formula can be obtained by applying a first-order Taylor series expansion to the entire logarithmized CES function at the point $\rho = 0$ (Uebe 2000). As the authors consider the latter approach as more straight-forward, the Kmenta approximation is called—in contrast to Kmenta (1967, p. 180)—first-order Taylor series expansion in the remainder of this paper.

The Kmenta approximation can be written also as a restricted translog function (Hoff 2004):

$$\begin{aligned}\log y = &\alpha_0 + \alpha_1 \log x_1 + \alpha_2 \log x_2 \\ &+ \frac{1}{2}\, \beta_{11}\, (\log x_1)^2 + \frac{1}{2}\, \beta_{22}\, (\log x_2)^2 + \beta_{12}\, \log x_1 \log x_2,\end{aligned} \tag{3}$$

where the two restrictions are

$$\beta_{12} = -\beta_{11} = -\beta_{22}. \tag{4}$$

If constant resturns to scale should be imposed, a third restriction

$$\alpha_1 + \alpha_2 = 1 \tag{5}$$

must be enforced. These restrictions can be utilised to test whether the linear Kmenta approximation of the CES (2) is an acceptable simplification of the translog functional form.[3] If this is the case, a simple t-test for the coefficient $\beta_{12} = -\beta_{11} = -\beta_{22}$ can be used to check if the Cobb-Douglas functional form is an acceptable simplification of the Kmenta approximation of the CES.[4]

The parameters of the CES function can be calculated from the parameters of the restricted translog function by

$$\gamma = \exp(\alpha_0) \tag{6}$$

$$\nu = \alpha_1 + \alpha_2 \tag{7}$$

$$\delta = \frac{\alpha_1}{\alpha_1 + \alpha_2} \tag{8}$$

[3] Note that this test does *not* check whether the *non-linear* CES function (1) is an acceptable simplification of the translog functional form or whether the *non-linear* CES function can be approximated by the Kmenta approximation.

[4] Note that this test does *not* compare the Cobb-Douglas function with the (non-linear) CES function but only with its linear approximation.

$$\rho = \frac{\beta_{12} \left(\alpha_1 + \alpha_2 \right)}{\alpha_1 * \alpha_2} \tag{9}$$

The Kmenta approximation of the CES function can be estimated by the function `cesEst`, which is included in the **micEconCES** package. If argument `method` of this function is set to `"Kmenta"`, it (a) estimates an unrestricted translog function (3), (b) carries out a Wald test of the parameter restrictions defined in equation (4) and eventually also in equation (5) using the (finite sample) F statistic, (c) estimates the restricted translog function (3, 4), and finally, (d) calculates the parameters of the CES using equations (6−9) as well as their covariance matrix using the delta method.

The following code estimates a CES function with the endogenous variable `y` (specified in argument `yName`), the two explanatory variables `x1` and `x2` (argument `xNames`), the artificial data set `cesData` that we generated above (argument `data`) using the Kmenta approximation (argument `method`) and allowing for variable returns to scale (argument `vrs`).

```
> cesKmenta <- cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData,
+     method = "Kmenta", vrs = TRUE)
```

Summary results can be obtained applying the `summary` method to the returned object.

```
> summary(cesKmenta)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "Kmenta")

Estimation by the linear Kmenta approximation
Test of the null hypothesis that the restrictions of the Translog
function required by the Kmenta approximation are true:
P-value = 0.6135929

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  0.74252    0.11009   6.745 1.53e-11 ***
delta  0.60864    0.03373  18.043  < 2e-16 ***
rho    0.71527    0.31722   2.255   0.0241 *
nu     1.21865    0.06617  18.416  < 2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.481868
Multiple R-squared: 0.7643218
```

The Wald test indicates that the restrictions on the Translog function implied by the Kmenta approximation cannot be rejected at any reasonable significance level.

To see whether the underlying technology is of the Cobb-Douglas form, we can check if the coefficient $\beta_{12} = -\beta_{11} = -\beta_{22}$ significantly differs from zero. As the estimation of the Kmenta approximation is stored in component `kmenta` of the object returned by `cesEst`, we can obtain summary information on the estimated coefficients of the Kmenta approximation by

```
> coef(summary(cesKmenta$kmenta))
```

```
                 Estimate Std. Error   t value   Pr(>|t|)
eq1_(Intercept) -0.2977003 0.14826207 -2.007933 0.04602347
eq1_a_1          0.7417197 0.05337124 13.897367 0.00000000
eq1_a_2          0.4769324 0.05156227  9.249638 0.00000000
eq1_b_1_1       -0.2076294 0.08907193 -2.331030 0.02076840
eq1_b_1_2        0.2076294 0.08907193  2.331030 0.02076840
eq1_b_2_2       -0.2076294 0.08907193 -2.331030 0.02076840
```

Given that $\beta_{12} = -\beta_{11} = -\beta_{22}$ significantly differs from zero at the 5% level, we can conclude that the underlying technology is not of the Cobb-Douglas form. Alternatively, we can check if the parameter ρ of the CES, which is calculated from the coefficients of the Kmenta approximation, significantly differs from zero. This should—as in our case—deliver similar results (see above).

Finally, we plot the fitted values against the actual endogenous variable (y) to check whether the parameter estimates are reasonable.

```
> compPlot(cesData$y, fitted(cesKmenta), xlab = "actual values",
+     ylab = "fitted values")
```

Figure 1 shows that the parameters produce reasonable fitted values.

However, the Kmenta approximation encounters several problems. First, it is a truncated Taylor series, whose remainder term must be seen as an omitted variable. Second, the Kmenta approximation converges to the underlying CES function only in a region of convergence, that is depending of the true parameters of the CES function (Thursby and Lovell 1978).

Although, Maddala and Kadane (1967) and Thursby and Lovell (1978) find estimates for ν and δ with small bias and MSE, results for γ and ρ are estimated with generally large bias and

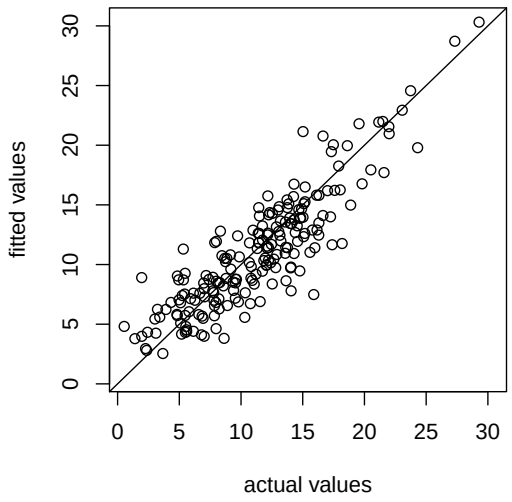

Figure 1: Fittet values from the Kmenta approximation against `y`

MSE (Thursby and Lovell 1978; Thursby 1980). More reliable results can only be obtained if $\rho \to 0$, and thus, $\sigma \to 1$ which increases the convergence region, i.e. if the underlying CES is of the Cobb-Douglas form. This is a major drawback of the Kmenta approximation as its purpose is to facilitate the estimation of functions with non-unitary σ.

2.2. Levenberg-Marquard algorithm

Initially, the Levenberg-Marquardt algorithm (Marquardt 1963) was most commonly used for estimating the parameters of the CES function by non-linear least-squares. This iterative algorithm is done by using an optimum interpolation between the Gauss-Newton method that involves a linearisation by a first-order Taylor series approximation and the gradient method (steepest-descent method).

In a Monte Carlo study by Thursby (1980) the Levenberg-Marquardt algorithm outperforms the other methods and gives the best estimates of the CES parameters. However, the Levenberg-Marquardt algorithm performs as poorly as the other methods in estimating the elasticity of substitution (σ), meaning that the estimated σ tends to be biased towards infinity, unity, or zero.

Although the Levenberg-Marquardt algorithm does not live up to modern standards, we include it for reasons of completeness, as it is has proven to be a standard method to estimate the CES technology. The Levenberg-Marquardt algorithm can be seen as a maximum neighbourhood method which performs an optimum interpolation between a first-order Taylor approximation (Gauss-Newton) and a steepest descend method (gradient method) (Marquardt 1963). By combining these two non-linear optimization algorithms, the developers want to increase conversion probability by reducing the weaknesses of each of the two methods.

The objective function $\Phi = ||Y - \hat{Y}||^2$ of a non-linear least-squares estimation does not fulfill the theoretical criteria of a well behaved function, unless the function value is close to its minimum. This feature becomes the more severe the more the function is non-linear. Therefore, it is crucial to find starting values close to the minimum. However, this is not always possible in practice. Choosing non-optimal starting values, the Gauss-Newton as well as the steepest descend method show a tendency to failure to convergence. If the starting values are too far from the minimum, the Gauss-Newton algorithm has difficulties to determine an appropriate step size, which can lead to step sizes either too big (cutting across the minimum) or too small (slow rates of convergence). On the other hand, the steepest descent method can handle suboptimal starting values very well, but shows a failure to convergence mostly due to very slow convergence when it gets close to the minimum (Kelley 1999).

In contrast to the Gauss-Newton and the steepest descend algorithms, the Levenberg-Marquardt algorithm determines the direction and the step size simultaneously, and thus, the algorithm proves to be more robust with higher rates of convergence, even if starting values are not optimal. If the Levenberg-Marquardt parameter λ is set to zero the algorithm turns to Gauss-Newton, for $\lambda \to \infty$ on the other hand it turns to steepest descent. Hence, as λ is defined in every iteration, the Levenberg-Marquardt algorithm uses the good global properties of the steepest descent method and—by approaching the minimum of the objective function—recovers the Gauss-Newton's fast convergence for small residual problems.

In the following we will give a rough outline of the algorithm.[5] We let

$$\langle Y_i(\boldsymbol{X}_i, \boldsymbol{\beta} + \boldsymbol{\gamma}) \rangle = f(\boldsymbol{X}_i, \boldsymbol{\beta}) + \sum_{j=1}^{k} \left(\frac{\partial f_i}{\partial \beta_j} \right) \gamma_j, \tag{10}$$

or shorter

$$\langle \boldsymbol{Y} \rangle = f_0 + P\gamma, \tag{11}$$

be the first-order Taylor series approximation, where Y_i is the ith value of the dependend variable, here output, $\boldsymbol{X}_i$ is the ith vector of covariates, $\boldsymbol{\beta}$ is a vector of parameters to be estimated, $\boldsymbol{\gamma}$ is a vector of small correction parameters to $\boldsymbol{\beta}$ calculated from a Taylor series with jth element γ_j, f is a differentiable function, k is the number of parameters to be estimated, f_0 is a vector of the first terms of the Taylor series, and $P = \partial f / \partial \beta$ is a Jacobian matrix. Then $\boldsymbol{\gamma}$ can be found by

$$(A + \lambda I)\boldsymbol{\gamma} = \boldsymbol{g}, \tag{12}$$

where I is an identity matrix and

$$A = P^\top P \tag{13}$$

[5] For a more detailed introduction into the Levenberg-Marquardt algorithm see Marquardt (1963) or Soda and Vichi (1976).

and

$$\boldsymbol{g} = P^\top (\boldsymbol{Y} - f_0). \tag{14}$$

Finally, let

$$\Phi(\gamma) = ||\boldsymbol{Y} - f_0 - P\gamma||^2 \tag{15}$$

The algorithm is then as follows: marquardt (μ, λ, Φ, r)

1. Let $\mu > 1$ be a tolerance parameter
2. Let $\lambda^{(r-1)}$ be the value from the previous iteration. Initially let $\lambda^{(0)} = 10^{-2}$.
3. Compute $\Phi(\lambda^{(r-1)})$ and $\Phi(\lambda^{(r-1)}/\mu)$
   (a) if $\Phi(\lambda^{(r-1)}/\mu) \leq \Phi^{(r)}$, let $\lambda^{(r)} = \lambda^{(r-1)}/\mu$.
   (b) if $\Phi(\lambda^{(r-1)}/\mu) > \Phi^{(r)}$, and $\Phi(\lambda^{(r-1)}) \leq \Phi^{(0)}$, let $\lambda^{(r)} = \lambda^{(r-1)}$.
   (c) if $\Phi(\lambda^{(r-1)}/\mu) > \Phi^{(r)}$, and $\Phi(\lambda^{(r-1)}) > \Phi^{(0)}$, increase λ by successive multiplication by μ until for some smallest w $\Phi(\lambda^{(r-1)}\mu^w) \leq \Phi^{(r)}$. Let $\lambda^{(r)} = \lambda^{(r-1)}\mu^w$.

To estimate a CES function by non-linear least-squares using the Levenberg-Marquardt algorithm, one can call the `cesEst` function with argument `method` set to `"LM"` or without this argument, as the Levenberg-Marquardt algorithm is the default estimation method used by `cesEst`. The user can modify a few details of this algorithm (e.g. different criterions for convergence) by adding argument `control` as described in the documentation of `nls.lm.control`. Argument `start` can be used to specify a vecor of starting values, where the order must be γ, δ, ρ (only if ρ is not fixed, e.g. during grid search), and ν (only if the model has variable returns to scale). If no starting values are provided, they are determined automatically (see section 3.7). We estimate the same example as before now by the Levenberg-Marquardt algorithm.

```
> cesLm <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE)
> summary(cesLm)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE)

Estimation by non-linear least-squares using the 'LM' optimizer
Convergence achieved after 4 iterations
Message: Relative error in the sum of squares is at most `ftol'.
```

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

Finally we plot the fitted values against the actual values `y` to see whether the estimated parameters are reasonable. The result is presented in figure 2.

```
> compPlot(cesData$y, fitted(cesLm), xlab = "actual values", ylab = "fitted values")
```

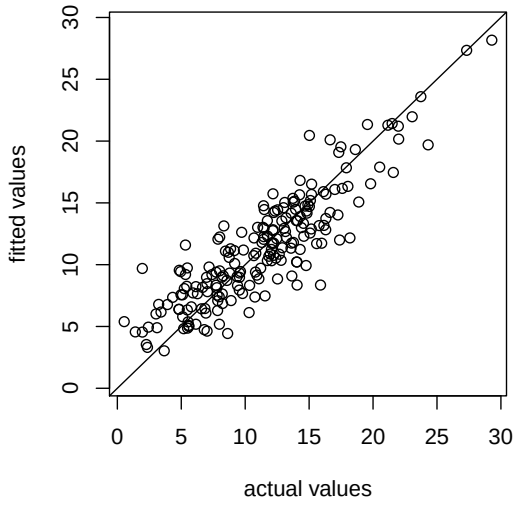

Figure 2: Fitted values from the LM algorithm against `y`

2.3. Alternative gradient-based optimisation algorithms

Several further gradient-based optimization algorithms that are suitable for non-linear least-squares estimations are implemented in R. Function `cesEst` can use some of them to estimate a CES function by non-linear least-squares. As a proper application of these estimation methods requires the user to be familiar with the main characteristics of the different algorithms, we will briefly discuss some practical issues of the algorithms that will be used to estimate the

CES function. However, it is not the aim of this paper to thoroughly discuss these algorithms. A detailed discussion of iterative optimisation algorithms is available, e.g., in Kelley (1999) or Mishra (2007).

One of the gradient-based optimization algorithms that can be used by `cesEst` is the "Conjugate Gradients" method based on Fletcher and Reeves (1964). This iterative method is mostly applied to optimization problems with many parameters and a large and possibly sparse Hessian matrix, because this algorithms requires neither storing nor inverting the Hessian matrix. The "Conjugated Gradient" method works best for objective functions that are approximately quadratic and it is sensitive to objective functions that are not well-behaved and have a non-positive semidefinite Hessian, i.e. convergence within the given number of iterations is less likely the more the level surface of the objective function differs from spherical (Kelley 1999). Given that the CES function has only few parameters and the objective function is not approximately quadratic and shows a tendency to "flat surfaces" around the minimum, the "Conjugated Gradient" method is probably less suitable than other algorithms for estimating a CES function. Setting argument `method` of `cesEst` to `"CG"` selects the "Conjugate Gradients" method for estimating the CES function by non-linear least-squares. The user can modify this algorithm (e.g. replacing the update formula of Fletcher and Reeves (1964) by the formula of Polak and Ribière (1969) or the one based on Sorenson (1969) and Beale (1972)) or some details (e.g. convergence tolerance level) by adding a further argument `control` as described in the "Details" section of the documentation of `optim`.

```
> cesCg <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "CG")
> summary(cesCg)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "CG")

Estimation by non-linear least-squares using the 'CG' optimizer
Convergence NOT achieved after 406 function and 101 gradient calls

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  0.99847    0.11124   8.976   <2e-16 ***
delta  0.62574    0.02807  22.294   <2e-16 ***
rho    0.60680    0.29187   2.079   0.0376 *
nu     1.09985    0.04503  24.427   <2e-16 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424191
Multiple R-squared: 0.7751486
```

Although the estimated parameters are similar to the estimates from the Levenberg-Marquardt algorithm, the “Conjugated Gradient” algorithm reports that it did not converge. Increasing the maximum number of iterations and the tolerance level leads to convergence. This indicates a slow convergence of the Conjugate Gradients algorithm for estimating the CES function.

```
> cesCg2 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "CG",
+     control = list(maxit = 1000, reltol = 1e-05))
> summary(cesCg2)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "CG", control = list(maxit = 1000, reltol = 1e-05))

Estimation by non-linear least-squares using the 'CG' optimizer
Convergence achieved after 1559 function and 387 gradient calls

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

Another algorithm supported by `cesEst` that is probably more suitable for estimating a CES function is an improved Newton-type method. As the original Newton method, this algorithm uses first and second derivatives of the objective function to determine the direction

of the shift vector and searches for a stationary point until the gradients are (almost) zero. However, in contrast to the original Newton method, this algorithm does a line search at each iteration to determine the optimal length of the shift vector (step size) as described in Dennis and Schnabel (1983) and Schnabel, Koontz, and Weiss (1985). Setting argument `method` of `cesEst` to `"Newton"` selects this improved Newton-type method. The user can modify a few details of this algorithm (e.g. the maximum step length) by adding further arguments that are described in the documentation of `nlm`. The following commands estimate a CES function by non-linear least-squares using this algorithm and print summary results.

```
> cesNewton <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
+     method = "Newton")
> summary(cesNewton)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "Newton")

Estimation by non-linear least-squares using the 'Newton' optimizer
Convergence achieved after 27 iterations

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

Furthermore, a quasi-Newton method developed independently by Broyden (1970), Fletcher (1970), Goldfarb (1970), and Shanno (1970) can be used by `cesEst`. This so-called BFGS algorithm also uses first and second derivatives and searches for a stationary point of the objective function where the gradients are (almost) zero. In contrast to the original Newton method, the BFGS method does a line search for the best step size and uses a special procedure to approximate and update the Hessian matrix in every iteration. The problem with

BFGS can be that although the current parameters are close to the minimum, the algorithm does not converge because the Hessian matrix at the current parameters is not close to the Hessian matrix at the minimum. However, in practice BFGS proves robust convergence (often superlinear) (Kelley 1999). If argument `method` of `cesEst` is `"BFGS"`, the BFGS algorithm is used for the estimation. The user can modify a few details of the BFGS algorithm (e.g. the convergence tolerance level) by adding the further argument `control` as described in the "Details" section of the documentation of `optim`.

```
> cesBfgs <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "BFGS")
> summary(cesBfgs)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "BFGS")

Estimation by non-linear least-squares using the 'BFGS' optimizer
Convergence achieved after 71 function and 15 gradient calls

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

2.4. Global optimization algorithms

While the gradient-based (local) optimization algorithms described above are designed to find local minima, global optimization algorithms, which are also known as direct search methods, are designed to find the global minimum. They are more tolerant to not well-behaved objective functions but they usually converge more slowly than the gradient-based methods. However, increasing computing power has made these algorithms suitable for day-to-day use.

One of these global optimization routines is the so-called Nelder-Mead algorithm (Nelder and Mead 1965), which is a downhill simplex algorithm. In every iteration $n+1$ vertices are defined in the n-dimensional parameter space. The algorithm converges by successively replacing the "worst" point by a new vertice in the n-dimensional parameter space. The Nelder-Mead algorithm has the advantage of a simple and robust algorithm, and is especially suitable for residual problems with non-differentiable objective functions. However, the heuristic nature of the algorithm causes slow convergence, especially close to the minimum, and can lead to convergence to non-stationary points. As the CES function is easily twice differentiable the advantage of the Nelder-Mead algorithm reduces to its robustness. As a consequence of the heuristic optimisation technique the results should be handled with care. However, the Nelder-Mead algorithm is much faster than the other global optimization algorithms described below. Function `cesEst` estimates a CES function with the Nelder-Mead algorithm if argument `method` is set to `"NM"`. The user can tweak this algorithm (e.g. the reflection factor, contraction factor, or expansion factor) or change some details (e.g. convergence tolerance level) by adding a further argument `control` as described in the "Details" section of the documentation of `optim`.

```
> cesNm <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "NM")
> summary(cesNm)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "NM")

Estimation by non-linear least-squares using the 'Nelder-Mead' optimizer
Convergence achieved after 359 iterations

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01024    0.11244   8.984   <2e-16 ***
delta  0.62710    0.02834  22.126   <2e-16 ***
rho    0.63961    0.29703   2.153   0.0313 *
nu     1.09544    0.04499  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
```

```
Multiple R-squared: 0.7751686
```

The Simulated Annealing algorithm was initially proposed by Kirkpatrick, Gelatt, and Vecchi (1983) and Cerny (1985) and is a modification of the Metropolis-Hastings algorithm. Every iteration chooses a random solution close to the current solution, while the probability of the choice is driven by a global parameter T which decreases as the algorithm moves on. Unlike other iterative optimisation algorithms, Simulated Annealing also allows T to increase which makes it possible to leave local minima. Therefore, Simulated Annealing is a robust global optimiser and can be applied to a large search space, where it provides fast and reliable solutions. Setting argument `method` to `"SANN"` selects a variant of the "Simulated Annealing" algorithm given in Bélisle (1992). The user can modify some details of the "Simulated Annealing" algorithm (e.g. the starting temperature or the number of function evaluations at each temperature) by adding a further argument `control` as described in the "Details" section of the documentation of `optim`. The only criterion for stopping this iterative process is the number of iterations and it does not indicate whether it converged or not.

```
> cesSann <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "SANN")
> summary(cesSann)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "SANN")

Estimation by non-linear least-squares using the 'SANN' optimizer

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  0.98474    0.10942   9.000   <2e-16 ***
delta  0.63192    0.02822  22.395   <2e-16 ***
rho    0.67107    0.29777   2.254   0.0242 *
nu     1.10819    0.04488  24.693   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.425541
Multiple R-squared: 0.774898
```

As the Simulated Annealing algorithm makes use of random numbers, the solution generally

depends on the initial "state" of R's random number generator. To ensure replicability, `cesEst` "seeds" the random number generator before it starts the "Simulated Annealing" algorithm with the value of argument `random.seed`, which defaults to 123. Hence, the estimation of the same model using this algorithm always returns the same estimates as long as argument `random.seed` is not altered (at least using the same software and hardware components).

```
> cesSann2 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "SANN")
> all.equal(cesSann, cesSann2)
```

```
[1] TRUE
```

It is recommended to start this algorithm with different values of argument `random.seed` and check whether the estimates differ considerably.

```
> cesSann3 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "SANN",
+     random.seed = 1234)
> cesSann4 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "SANN",
+     random.seed = 12345)
> cesSann5 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "SANN",
+     random.seed = 123456)
> m <- rbind(cesSann = coef(cesSann), cesSann3 = coef(cesSann3),
+     cesSann4 = coef(cesSann4), cesSann5 = coef(cesSann5))
> rbind(m, stdDev = sd(m))
```

```
              gamma       delta        rho         nu
cesSann  0.98473949 0.631924860 0.67107287 1.10818592
cesSann3 1.03244315 0.638666280 0.80004041 1.08781915
cesSann4 1.09853191 0.640687668 0.74022400 1.06285986
cesSann5 1.03797816 0.633285471 0.63724351 1.08296884
stdDev   0.04665818 0.004202345 0.07259749 0.01861008
```

If the estimates differ remarkably, the user might try to increase the number of iterations, which is 10,000 by default. Now we re-estimate the model a few times with 100,000 iterations each.

```
> cesSannB <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "SANN",
+     control = list(maxit = 1e+05))
> cesSannB3 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
+     method = "SANN", random.seed = 1234, control = list(maxit = 1e+05))
> cesSannB4 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
```

```
+     method = "SANN", random.seed = 12345, control = list(maxit = 1e+05))
> cesSannB5 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
+     method = "SANN", random.seed = 123456, control = list(maxit = 1e+05))
> m <- rbind(cesSannB = coef(cesSannB), cesSannB3 = coef(cesSannB3),
+     cesSannB4 = coef(cesSannB4), cesSannB5 = coef(cesSannB5))
> rbind(m, stdDev = sd(m))

                gamma       delta        rho          nu
cesSannB  1.019018933 0.626396980 0.62656297 1.091731388
cesSannB3 1.012203547 0.629279618 0.66705294 1.094746865
cesSannB4 1.017438463 0.630137829 0.65935962 1.092449539
cesSannB5 1.000831695 0.634353090 0.64772478 1.099822550
stdDev    0.008227146 0.003289482 0.01763287 0.003656927
```

The estimates are much more similar now—only the estimates of ρ still differ somewhat.

In contrary to the other algorithms described in this paper, the Differential Evolution algorithm (Storn and Price 1997) belongs to the class of evolution strategy optimisers and convergence cannot be proven analytically. However, the algorithm has proven to be effective and accurate on a large range of optimisation problems, inter alia the CES function (Mishra 2007). For some problems it has proven to be more accurate and more efficient than Simulated Annealing, Quasi-Newton, or other genetic algorithms (Storn and Price 1997; Ali and Törn 2004; Mishra 2007). Function `cesEst` uses a Differential Evolution optimizer for the non-linear least-squares estimation of the CES function, if argument `method` is set to `"DE"`. The user can modify the Differential Evolution algorithm (e.g. the differential evolution strategy or selection method) or change some details (e.g. the number of population members) by adding a further argument `control` as described in the documentation of `DEoptim.control`. In contrary to the other ompimisation algorithms, the Differential Evolution method requires finite boundaries of the parameters. By default, the bounds are $0 \leq \gamma \leq 10^{10}$, $0 \leq \delta \leq 1$, $-1 \leq \rho \leq 10$, and $0 \leq \nu \leq 10$. Of course, the user can specify own lower and upper bounds by setting arguments `lower` and `upper` to numeric vectors.

```
> cesDe <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     control = list(trace = FALSE))
> summary(cesDe)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
```

```
    method = "DE", control = list(trace = FALSE))

Estimation by non-linear least-squares using the 'DE' optimizer

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01012    0.11256   8.974   <2e-16 ***
delta  0.62777    0.02841  22.094   <2e-16 ***
rho    0.64216    0.29785   2.156   0.0311 *
nu     1.09500    0.04505  24.307   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.424131
Multiple R-squared: 0.7751597
```

Likewise the "Simulated Annealing" algorithm, the Differential Evolution algorithm makes use of random numbers and `cesEst` "seeds" the random number generator with the value of argument `random.seed` before it starts this algorithm to ensure replicability.

```
> cesDe2 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     control = list(trace = FALSE))
> all.equal(cesDe, cesDe2)

[1] TRUE
```

It is recommended also for this algorithm to check if different values of argument `random.seed` result in remarkably different estimates.

```
> cesDe3 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     random.seed = 1234, control = list(trace = FALSE))
> cesDe4 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     random.seed = 12345, control = list(trace = FALSE))
> cesDe5 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     random.seed = 123456, control = list(trace = FALSE))
> m <- rbind(cesDe = coef(cesDe), cesDe3 = coef(cesDe3), cesDe4 = coef(cesDe4),
+     cesDe5 = coef(cesDe5))
> rbind(m, stdDev = sd(m))

             gamma        delta         rho          nu
cesDe  1.010119217 0.6277687357 0.642157064 1.095001892
```

```
cesDe3 1.008841004 0.6287090120 0.637607137 1.096142323
cesDe4 1.011146256 0.6273823595 0.641914728 1.094396769
cesDe5 1.000593563 0.6268241309 0.640145999 1.098865426
stdDev 0.004814219 0.0007932049 0.002100288 0.001979579
```

These estimates are rather similar, which generally indicates that all estimates are close to the optimum (minimum of the sum of squared residuals). However, if the user wants to get more precise estimates than obtained with the default settings of this algorithm, e.g. if the estimates differ considerably, the user might try to increase the number of iterations, which is 200 by default. Now we re-estimate this model a few times with 1,000 iterations each.

```
> cesDeB <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     control = list(trace = FALSE, itermax = 1000))
> cesDeB3 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     random.seed = 1234, control = list(trace = FALSE, itermax = 1000))
> cesDeB4 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     random.seed = 12345, control = list(trace = FALSE, itermax = 1000))
> cesDeB5 <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "DE",
+     random.seed = 123456, control = list(trace = FALSE, itermax = 1000))
> rbind(cesDeB = coef(cesDeB), cesDeB3 = coef(cesDeB3), cesDeB4 = coef(cesDeB4),
+     cesDeB5 = coef(cesDeB5))

         gamma     delta       rho       nu
cesDeB  1.0102 0.6271122 0.6397527 1.095452
cesDeB3 1.0102 0.6271122 0.6397527 1.095452
cesDeB4 1.0102 0.6271122 0.6397527 1.095452
cesDeB5 1.0102 0.6271122 0.6397527 1.095452
```

The estimates are virtually identical now.

2.5. Constraint parameters

As a meaningful analysis based on a CES function requires that this function is consistent with economic theory, it is often desirable to constrain the parameter space to the economically meaningful region.

Function `cesEst` can estimate a CES function under parameter constraints using a modification of the BFGS algorithm suggested by Byrd, Lu, Nocedal, and Zhu (1995). In contrary to the ordinary BFGS algorithm summarized above, the so-called L-BFGS-B algorithm allows for box-constraints on the parameters and also does not explicitly form or store the Hessian matrix but instead relies on the past (often less than 10) values of the parameters and the

gradient vector. Therefore, the L-BFGS-B algorithm is especially suitable for high dimensional optimisation problems but—of course—it can be also used for optimisation problems with only a few parameters (as the CES function). Function `cesEst` estimates a CES function with parameter constraints using the L-BFGS-B algorithm if argument `method` is set to `"L-BFGS-B"`. The user can tweak some details of this algorithm (e.g. the number of BFGS updates) by adding a further argument `control` as described in the "Details" section of the documentation of `optim`. By default, the restrictions on the parameters are $0 \leq \gamma \leq \infty$, $0 \leq \delta \leq 1$, $-1 \leq \rho \leq \infty$, and $0 \leq \nu \leq \infty$. The user can specify own lower and upper bounds by setting arguments `lower` and `upper` to numeric vectors.

```
> cesLbfgsb <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
+     method = "L-BFGS-B")
> summary(cesLbfgsb)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "L-BFGS-B")

Estimation by non-linear least-squares using the 'L-BFGS-B' optimizer
Convergence achieved after 36 function and 36 gradient calls
Message: CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

The so-called PORT routines (Gay 1990) include a quasi-Newton optimisation algorithm that allows for box constraints on the parameters and has several advantages over traditional Newton routines, e.g. trust regions and reverse communication. Setting argument `method` to

`"PORT"` selects a the optimisation algorithm of the PORT routines. The user can modify a few details of the Newton algorithm (e.g. the minimum step size) by adding a further argument `control` as described in section "Control parameters" of the documentation of `nlminb`. The lower and upper bounds of the parameters have the same default values as for the L-BFGS-B method.

```
> cesPort <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, method = "PORT")
> summary(cesPort)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "PORT")

Estimation by non-linear least-squares using the 'PORT' optimizer
Convergence achieved after 27 iterations
Message: relative convergence (4)

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

2.6. Grid search for ρ

As the objective function for estimating the CES by non-linear least-squares shows a tendency to "flat surfaces" around the minimum—in particular for a wide range of values for ρ— many optimization algorithms have problems in finding the minimum of the objective function. This problem can be alleviated by performing a one-dimensional grid search, where a sequence of values for ρ is pre-selected and the remaining parameters are estimated by non-linear least-squares holding ρ fixed at each of the pre-defined values. Later, the estimation with the value of ρ that results in the smallest sum of squared residuals is chosen.

The function `cesEst` carries out this grid search procedure, if the user sets its argument `rho` to a numeric vector containing the values of ρ that should be used in the grid search. The estimation of the other parameters during the grid search can use all non-linear optimization algorithms described above. Since the "best" value of ρ that was found in the grid search is not known but estimated (as the other parameters but with a different method), the covariance matrix of the estimated parameters includes ρ and is calculated as if ρ was estimated as usual. The following command estimates the CES function by a one-dimensional grid search for ρ, where the pre-selected values for ρ are the values from -0.3 to 1.5 with an increment of 0.1 and the default optimisation method, the Levenberg-Marquardt algorithm is used to estimate the remaining parameters.

```
> cesGrid <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, rho = seq(from = -0.3,
+     to = 1.5, by = 0.1))
> summary(cesGrid)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    rho = seq(from = -0.3, to = 1.5, by = 0.1))

Estimation by non-linear least-squares using the 'LM' optimizer
and a one-dimensional grid search for coefficient 'rho'
Convergence achieved after 4 iterations
Message: Relative error in the sum of squares is at most `ftol'.

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.00527    0.11192   8.982   <2e-16 ***
delta  0.62568    0.02809  22.272   <2e-16 ***
rho    0.60000    0.29151   2.058   0.0396 *
nu     1.09699    0.04501  24.374   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.424194
Multiple R-squared: 0.775148
```

An overview of the relationship between the pre-selected values of ρ and the corresponding

sums of the squared residuals can be obtained by applying the `plot` method.[6]

```
> plot(cesGrid)
```

This overview is shown in figure 3.

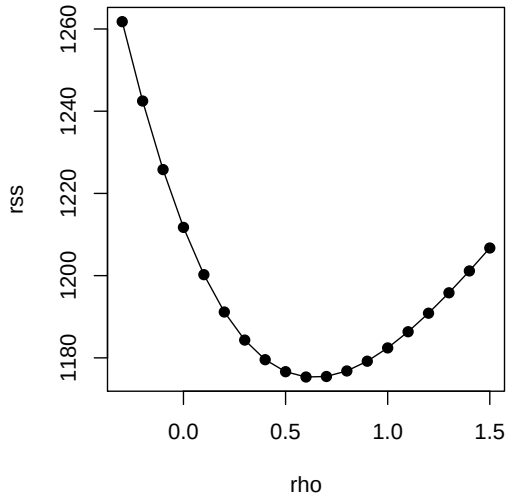

Figure 3: Values of ρ and corresponding sums of squared residuals

The results of this grid search algorithm can be either used directly or used as starting values for a non-linear least-squares estimation so that also ρ values between the grid points can be estimated. Starting values can be set by argument `startVal`.

```
> cesStartGrid <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
+     start = coef(cesGrid))
> summary(cesStartGrid)

Estimated CES function with variable returns to scale

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    start = coef(cesGrid))

Estimation by non-linear least-squares using the 'LM' optimizer
Convergence achieved after 3 iterations
Message: Relative error in the sum of squares is at most `ftol'.
```

[6] This `plot` method can be applied only if the model was estimated by grid search.

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
gamma  1.01020    0.11244   8.984   <2e-16 ***
delta  0.62711    0.02834  22.126   <2e-16 ***
rho    0.63975    0.29705   2.154   0.0313 *
nu     1.09545    0.04500  24.346   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.424083
Multiple R-squared: 0.7751686
```

3. Implementation

The function `cesEst` is the primary user interface of the **micEconCES** package (Henningsen and Henningsen 2010). However, the actual estimations are carried out by internal helper functions or functions from other packages.

3.1. Kmenta approximation

The estimation of the Kmenta approximation (2) is implemented in the internal function `cesEstKmenta`. This function uses `translogEst` from the **micEcon** package (Henningsen 2009) for estimating the unrestricted translog function (3). The test of the parameter restrictions defined in equation (4) is performed by the function `linear.hypothesis` of the **car** package (Fox 2009). The restricted translog model (3, 4) is estimated with function `systemfit` from the **systemfit** package (Henningsen and Hamann 2007).

3.2. Non-linear least-squares estimation

The non-linear least-squares estimations are carried out by various optimisers from other packages. Estimations with the Levenberg-Marquardt algorithm are performed by function `nls.lm` of the **minpack.lm** package (Elzhov and Mullen 2009), which is an R interface to the FORTRAN package **MINPACK** (Moré, Garbow, and Hillstrom 1980). Estimations with the Conjugate Gradients (CG), BFGS, Nelder-Mead (NM), Simulated Annealing (SANN), and L-BFGS-B algorithms use the function `optim` from the **stats** package (R Development Core Team 2009). Estimations with the Newton-type algorithm are performed by function `nlm` from the **stats** package (R Development Core Team 2009), which uses the FORTRAN library **UNCMIN** (Schnabel *et al.* 1985) with line search as step selection strategy. Estimations with the Differential Evolution (DE) algorithm are performed by function `DEoptim` from the

DEoptim package (Ardia and Mullen 2009). Estimations with the PORT routines use function `nlminb` from the **stats** package (R Development Core Team 2009), which uses the FORTRAN library **PORT** (Gay 1990).

3.3. Grid search

The grid search procedure is implemented in the internal function `cesEstGridRho`. This function consecutively calls `cesEst` for each of the pre-selected values of ρ, where argument `rho` of `cesEst` is set to one of the pre-selected values at each call. If argument `rho` of `cesEst` is a single scalar value, `cesEst` does not perform a grid search but estimates the CES function by non-linear least-squares with parameter ρ fixed at the value of argument `rho`.

3.4. Calculating output

Function `cesCalc` can be used to calculate the output quantity of the CES function given input quantities and parameters. An example of using `cesCalc` is shown in the beginning of section 2, where the output variable of an artificial data set that is used to demonstrate the usage of `cesEst` is generated with this function. Furthermore, the `cesCalc` function is used by the internal function `cesRss`, that calculates and returns the sum of squared residuals, which is the objective function in the non-linear least-squares estimations. As the CES function is not defined for $\rho = 0$, `cesCalc` calculates in this case the output quantity with the limit of the CES function for $\rho \to 0$, which is the Cobb-Douglas function.

We noticed that the calculations with `cesCalc` using equation (1) are imprecise when ρ is close to 0. This is caused by rounding errors that are unavoidable on digital computers but are usually negligible. However, rounding errors can get large in specific circumstances, e.g. in the CES function with very small ρ, when very small (in absolute terms) exponents ($-\rho$) are applied first and then a very large (in absolute terms) exponent ($-\nu/\rho$) is applied. Therefore, `cesCalc` uses a first-order Taylor series approximation at the point $\rho = 0$ for calculating the output of the CES function, if the absolute value of ρ is smaller than or equal to argument `rhoApprox`, which is $5 \cdot 10^{-6}$ by default. This first-order Taylor series approximation is the Kmenta approximation defined in (2). We illustrate this in the left panel of figure 4, which has been created by following commands.

```
> rhoData <- data.frame(rho = seq(-2e-06, 2e-06, 5e-09), yCES = NA,
+     yLin = NA)
> for (i in 1:nrow(rhoData)) {
+     cesCoef <- c(gamma = 1, delta = 0.6, rho = rhoData$rho[i],
+         nu = 1.1)
+     rhoData$yLin[i] <- cesCalc(xNames = c("x1", "x2"), data = cesData[1,
+         ], coef = cesCoef, rhoApprox = Inf)
```

```
+     rhoData$yCES[i] <- cesCalc(xNames = c("x1", "x2"), data = cesData[1,
+         ], coef = cesCoef, rhoApprox = 0)
+ }
> rhoData$yCES <- rhoData$yCES - rhoData$yLin[rhoData$rho == 0]
> rhoData$yLin <- rhoData$yLin - rhoData$yLin[rhoData$rho == 0]
> plot(rhoData$rho, rhoData$yCES, type = "l", col = "red", xlab = "rho",
+     ylab = "y (normalised, red = CES, black = linearised)")
> lines(rhoData$rho, rhoData$yLin)
```

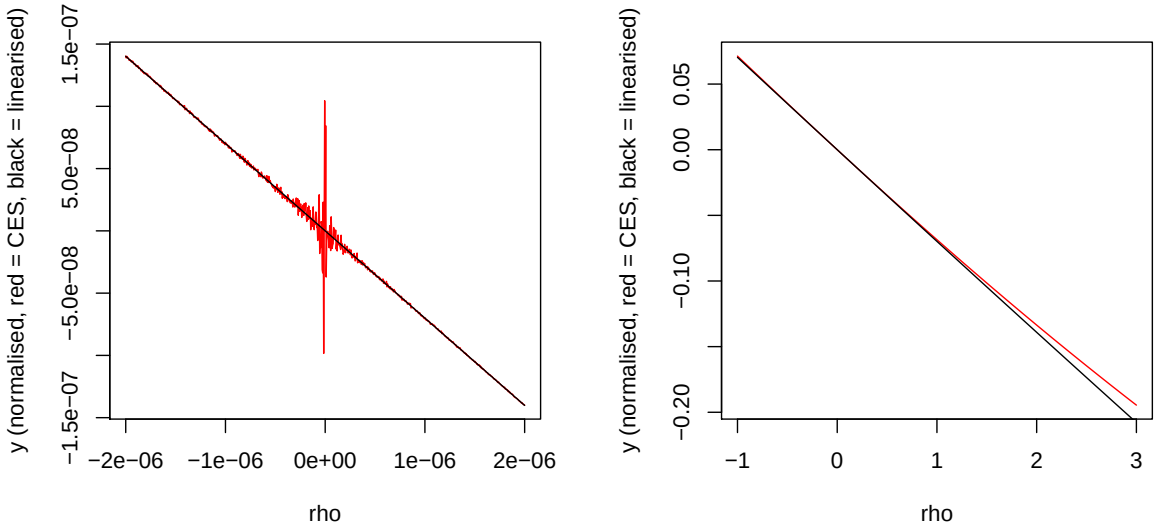

Figure 4: Calculated output for different values of ρ

The right panel of figure 4 shows that the relationship between ρ and the output y can be rather precisely approximated by a linear function, because it is nearly linear for a wide range of ρ values.[7]

When estimating a CES function with function `cesEst`, the user can use argument `rhoApprox` to modify the threshold for calculating the endogenous variable by the Kmenta approximation (2), as the first element of the vector `rhoApprox` is passed to `cesCalc`, partly through `cesRss`. This might affect not only the fitted values and residuals returned by `cesEst`, but also the estimation results, because the endogenous variable is used to calculate the sum of squared residuals, which is the objective function of the non-linear least-squares estimations.

3.5. Partial derivatives with respect to coefficients

The internal function `cesDerivCoef` returns the partial derivatives of the CES function with

[7] The commands for creating the right panel of figure 4 are not shown here, because they are the same as the commands for the left panel of this figure except for the command for creating the vector of ρ values.

respect to all coefficients at all provided data points. These partial derivatives are:

$$\frac{\partial y}{\partial \gamma} = \left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \tag{16}$$

$$\frac{\partial y}{\partial \delta} = -\frac{\gamma\,\nu}{\rho}\left(x_1^{-\rho} - x_2^{-\rho}\right)\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)^{-\frac{\nu}{\rho}-1} \tag{17}$$

$$\begin{aligned}\frac{\partial y}{\partial \rho} = {} & \frac{\gamma\,\nu}{\rho^2}\,\log\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \\ & + \frac{\gamma\,\nu}{\rho}\left(\delta \log(x_1) x_1^{-\rho} + (1-\delta)\log(x_2)x_2^{-\rho}\right)\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)^{-\frac{\nu}{\rho}-1}\end{aligned} \tag{18}$$

$$\frac{\partial y}{\partial \nu} = -\frac{\gamma}{\rho}\,\log\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \tag{19}$$

These derivatives are not defined for $\rho = 0$ and are imprecise if ρ is close to zero (similar to the output variable of the CES function, see section 3.4). Therefore, we calculate these derivatives by first-order Taylor series approximations at the point $\rho = 0$ if ρ is zero or close to zero:

$$\frac{\partial y}{\partial \gamma} = x_1^{\nu\,\delta}\; x_2^{\nu\,(1-\delta)} \exp\left(-\frac{\rho}{2}\,\nu\,\delta\,(1-\delta)\,(\log x_1 - \log x_2)^2\right) \tag{20}$$

$$\begin{aligned}\frac{\partial y}{\partial \delta} = {} & (\log x_1 - \log x_2)\, x_1^{\nu\,\delta}\; x_2^{\nu(1-\delta)} \\ & \left(1 - \frac{\rho}{2}\left[1 - 2\,\delta + \nu\,\delta(1-\delta)\,(\log x_1 - \log x_2)\,\right](\log x_1 - \log x_2)\right)\end{aligned} \tag{21}$$

$$\begin{aligned}\frac{\partial y}{\partial \rho} = {} & \gamma\,\nu\,\delta(1-\delta) x_1^{\nu\,\delta}\; x_2^{\nu(1-\delta)}\Big(-\frac{1}{2}\,(\log x_1 - \log x_2)^2 \\ & + \frac{\rho}{3}(1-2\,\delta)\,(\log x_1 - \log x_2)^3 + \frac{\rho}{4}\,\nu\,\delta(1-\delta)\,(\log x_1 - \log x_2)^4\Big)\end{aligned} \tag{22}$$

$$\begin{aligned}\frac{\partial y}{\partial \nu} = {} & \gamma\, x_1^{\nu\,\delta}\; x_2^{\nu(1-\delta)}\Big(\delta \log x_1 + (1-\delta)\log x_2 \\ & - \frac{\rho}{2}\,\delta(1-\delta)\,(\log x_1 - \log x_2)^2\,[1 + \nu\,(\delta \log x_1 + (1-\delta)\log x_2)]\Big)\end{aligned} \tag{23}$$

Function `cesDerivCoef` has an argument `rhoApprox` that can be used to set the threshold levels for defining when ρ is "close" to zero. This argument must be a numeric vector with exactly four elements that define the thresholds for $\partial y/\partial\gamma$, $\partial y/\partial\delta$, $\partial y/\partial\rho$, and $\partial y/\partial\nu$, respectively. By default, these thresholds are $5 \cdot 10^{-6}$ for $\partial y/\partial\gamma$, $5 \cdot 10^{-6}$ for $\partial y/\partial\delta$, 10^{-3} for $\partial y/\partial\rho$, and $5 \cdot 10^{-6}$ for $\partial y/\partial\nu$.

Function `cesDerivCoef` is used to provide argument `jac` to function `nls.lm` so that the Levenberg-Marquardt algorithm can use analytical derivatives of each residual with respect to the coefficients. Furthermore, this function is used by the internal function `cesRssDeriv`, which calculates the partial derivatives of the sum of squared residuals (RSS) with respect to

the coefficients by

$$\frac{\partial \text{RSS}}{\partial \theta} = -2 \sum_{i=1}^{N} \left(u_i \frac{\partial y_i}{\partial \theta} \right), \tag{24}$$

where N is the number of observations, u_i is the residual of the ith observation, $\theta \in \{\gamma, \delta, \rho, \nu\}$ is a coefficient of the CES function, and $\partial y_i / \partial \theta$ is the partial derivative of the CES function with respect to coefficient θ evaluated at the ith observation as defined in equations (16) to (19) or—depending on the value of ρ and argument `rhoApprox`—equations (20) to (23). Function `cesRssDeriv` is used to provide analytical gradients for the gradient-based optimization algorithms, i.e. Conjugate Gradients, Newton-type, BFGS, L-BFGS-B, and PORT. Finally, function `cesDerivCoef` is used to obtain the gradient matrix for calculating the asymptotic covariance matrix of the non-linear least-squares estimator (see section 3.6).

When estimating a CES function with function `cesEst`, the user can use argument `rhoApprox` to modify the thresholds for calculating the derivatives with respect to the coefficients by the linear approximations (20) to (23), as a vector containing the second to the fifth element of argument `rhoApprox` is passed to `cesDerivCoef`, partly through `cesRssDeriv`. This might affect not only the covariance matrix of the estimates, but also the estimation results obtained by a gradient-based optimisation algorithm.

3.6. Covariance matrix

The asymptotic covariance matrix of the non-linear least-squares estimator obtained by the various iterative optimisation methods is calculated by (Greene 2008, p. 292)

$$\hat{\sigma}^2 \left(\left(\frac{\partial y}{\partial \theta} \right)^\top \frac{\partial y}{\partial \theta} \right)^{-1}, \tag{25}$$

where $\partial y / \partial \theta$ denotes the $N \times k$ gradient matrix defined in equations (16) to (19), N is the number of observations, k is 3 for CES functions with constant returns to scale (ν not estimated but fixed at 1) and 4 for CES functions with variable returns to scale (ν estimated), and $\hat{\sigma}^2$ denotes the estimated variance of the residuals. As equation (25) is only valid asymptotically, we calculate the estimated variance of the residuals by

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{N} u_i^2, \tag{26}$$

i.e. without correcting for degrees of freedom.

3.7. Starting values

If the user calls `cesEst` with argument `start` set to a vector of starting values, the internal function `cesEstStart` checks if the number of sarting values is correct and if the individual

starting values are in the appropriate range of the corresponding parameter. If no starting values are provided by the user, function `cesEstStart` determines the starting values automatically. The starting value of δ is always set to 0.5. If the coefficient ρ is estimated (not fixed as, e.g., during grid search), the starting value of ρ is set to 0.25, which corresponds to an elasticity of substitution of 0.8. If the estimation allows for a model with variable returns to scale, the starting value of ν is set to 1, which corresponds to constant returns to scale. Finally, the starting value of γ is set to a value so that the mean of the endogenous variable is equal to the mean of its fitted values, i.e.

$$\gamma = \frac{\frac{1}{N}\sum_{i=1}^{N} y_i}{\frac{1}{N}\sum_{i=1}^{N}\left(0.5\, x_{1i}^{-\rho_0} + 0.5\, x_{2i}^{-\rho_0}\right)^{-\frac{1}{\rho_0}}}, \tag{27}$$

where ρ_0 is either the pre-selected value of ρ (if ρ is fixed) or the starting value of ρ, i.e. 0.25 (if ρ is estimated).

3.8. Other internal functions

The internal function `cesCoefAddRho` is used to add the value of ρ to the vector of coefficients, when ρ is fixed (e.g. during grid search for ρ) and hence, not included in the vector of estimated coefficients.

If the user selects the optimization algorithm Differential Evolution, L-BFGS-B, or PORT but does not specify lower or upper bounds of the coefficients, the internal function `cesCoefBounds` creates and returns the default bounds depending on the optimization algorithm as described in sections 2.4 and 2.5.

The internal function `cesCoefNames` returns a vector of character strings, which are the names of the coefficients of the CES function.

3.9. Methods

The **micEconCES** package makes use of the "S3" class system of the R language introduced in Chambers and Hastie (1992). Objects returned by function `cesEst` are of class `"cesEst"` and the **micEconCES** package includes several methods for objects of this class. The `print` method prints the call and the estimated coefficients. The `coef`, `vcov`, `fitted`, and `residuals` methods extract and return the estimated coefficients, their covariance matrix, the fitted values, and the residuals, respectively. The `plot` method can be applied only if the model was estimated by grid search; it plots a scatter plot of the pre-selected values of ρ against the corresponding sums of the squared residuals (see section 2.6) by using the `plot.default` and `points` commands of the **graphics** package (R Development Core Team 2009).

The `summary` method calculates the estimated standard error of the residuals ($\hat{\sigma}$), the cov-

ariance matrix of the coefficients estimated by non-linear least-squares, the R^2 value as well as the standard errors, t-values, and marginal significance levels (P values) of the estimated parameters. The object returned by the `summary` method is of class `"summary.cesEst"`. The `print` method for objects of class `"summary.cesEst"` prints the call, the estimated coefficients, their standard errors, t-values, and marginal significance levels as well as some information on the estimation procedure (e.g. algorithm, convergence). The `coef` method for objects of class `"summary.cesEst"` returns a matrix with four columns containing the estimated coefficients, their standard errors, t-values, and marginal significance levels, respectively.

4. Monte Carlo study

In this section we perform a Monte Carlo study to compare the different estimation methods described above. These are the estimation by R's standard tool for non-linear least-squares estimations, `nls`, as well as the linear estimation of the Kmenta approximation and the non-linear least-squares estimations using the various optimization algorithms described in sections 2.2 to 2.5 using function `cesEst`.[8] The data set used in this Monte Carlo study has 100 observations, where the input quantitites are drawn from a χ^2 distribution with 10 degrees of freedom. We generate the "deterministic" output quantity by a CES function with variable returns to scale, where the parameters are $\gamma = 1$, $\delta = 0.6$, $\rho = 1/3$, and $\nu = 1.1$. This function has an elasticity of substitution of $\sigma = 0.75$. In each of the 1000 replications, a new set of disturbance terms is drawn from a normal distribution with a mean of zero and standard deviation of 1.5. This results in R^2 values of the estimated models of around 0.915

Function `cesEst` is generally called with the default values of all arguments (except for argument `method`, of course). However, we override following default settings:

- Function `nls`:
  we set the control parameter `warnOnly` to `TRUE` so that this function returns coefficients (rather than just an error message) if the optimization does not converge.

- Levenberg-Marquardt, Newton, BFGS, L-BFGS-B:
  we increased the maximum number of iterations to 250 to increase the chance that these algorithms reach convergence.

- Conjugate Gradients:
  we changed control parameter `type` to 2 so that the update formula of Polak and Ribière (1969) is used, increased the maximum number of iterations to 1000 and increased the

[8] The estimation by `nls` in this Monte Carlo study is done through function `cesEst`, which uses `nls` for the estimation if argument `method` is set to `"nls"`. This feature is not mentioned in the documentation of `cesEst`, because it is not completely implemented yet.

tolerance level (argument `reltol`) to 10^{-4} so that this algorithm reaches convergence in most replications (see example in section 2.3)

- Simulated Annealing:
  we increased the number of iterations to $50,000$ so that the estimate is closer to the global minimum of the objective function (see section 2.4)
- Differential Evolution:
  we increased the number of iterations to $1,000$ so that the estimate is closer to the global minimum of the objective function (see section 2.4)

The script used for the Monte Carlo simulation is shown in appendix B. The general results of this Monte Carlo study are shown in table 1. Function `nls` reports 29 times that the non-linear minimization of the squared residuals has not converged. The Newton and the Nelder-Mead algorithms report this 5 and 3 times, respectively. All other algorithms always report convergence. Even if `nls` or the Newton or Nelder-Mead algorithm report non-convergence, the coefficients estimated by those methods are very close to the coefficients estimated by most other methods. Moreover, sum of squared residuals of the "non-converged" estimations is virtually the same as the sum of squared residuals of most other algorithms in the same replication. Hence, it seems that only the default values of the convergence tolerance of `nls` and the Newton and Nelder-Mead algorithms, which are used in this Monte Carlo study, are a little too low for this optimization problem. The average sums of the squared residuals are virtually identical for most estimation methods; only the Simulated Annealing method has on average slightly larger sums of the squared residuals and the Kmenta approximation, which does *not* aim at minimizing the sum of squared residuals of the (non-linear) CES function, has a somewhat larger average sum of squared residuals.

We summarize the results of the Monte Carlo study by presenting the biases and root mean square errors (RMSE) of the coefficients and the elasticity of substitution. The bias of a parameter θ estimated by method m is

$$\frac{1}{K}\sum_{i=1}^{K}\hat{\theta}_{i,m}-\theta, \tag{28}$$

where K is the number of replications in the Monte Carlo study, $\hat{\theta}_{i,m}$ is the estimate of parameter θ estimated by method m in the ith replication, and θ is the true value of this parameter. The root mean square error of this parameter θ estimated by method m is

$$\sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(\hat{\theta}_{i,m}-\theta\right)^2}, \tag{29}$$

where all variables are as defined above.

Table 1: General results of the Monte Carlo simulation

	nNoConv	nConv	rssAll	rssConv
Kmenta	0	1000	228.2826052	228.6947658
nls	29	971	216.7175798	217.0249514
LM	0	1000	216.7175777	217.0249514
CG	0	1000	216.7175778	217.0249514
Newton	5	995	216.7175779	217.0249515
BFGS	0	1000	216.7175823	217.0249561
Nelder-Mead	3	997	216.7176217	217.0249958
SANN	0	1000	216.9664636	217.2736183
DE	0	1000	216.7175777	217.0249514
L-BFGS-B	0	1000	216.7175782	217.0249518
PORT	0	1000	216.7175777	217.0249514

Description of columns:

nNoConv: number of replications, where the estimation procedure with the corresponding method warned about non-convergence

nConv: number of replications, where the estimation with the corresponding method converged

rssAll: mean sum of squared residuals of all replications

rssConv: mean sum of squared residuals of the replications, where all methods converged

The biases of the estimated coefficients of the CES function and of the elasticity of substitution determined in our Monte Carlo study are shown in table 2. These biases are generally very small, which means that the *means* of the estimated parameters are very close to their true values, no matter which estimation method is used. Only the Kmenta approximation returns on average a γ that is somewhat too small and a ν that is a little too large but the bias of δ and ρ is even smaller than the corresponding biases from the non-linear least-squares estimations. The estimated elasticities of substitution are on average a little larger than the true value—particularly for the Kmenta approximation.[9]

The root mean square errors (RMSE) of the estimated coefficients of the CES function and of the elasticity of substitution obtained by our Monte Carlo study are shown in table 3. The RMSEs of γ, δ, and ν are mostly rather small, which means that these coefficients are estimated rather precisely, i.e. the estimated coefficients are mostly very close to their true values. In contrast, the RMSEs of ρ are rather large, which means that the estimation of this coefficient is rather imprecise. However, the elasticities of substitution calculated from the estimated ρs have rather small RMSEs, i.e. are mostly rather close to their true values. As the elasticities of substitution—and not the ρs—are usually used for interpreting the substitutability of inputs, the imprecise estimation of ρ is not a major problem. The RMSEs of most algorithms for non-linear least-squares estimations are virtually identical. The RMSEs of the Simulated Annealing algorithm are slightly larger than the RMSEs of the

[9] This is a little surprising as the Kmenta approximation has the smallest bias of the corresponding parameter ρ but this can be explained by the non-linear relationship between ρ and σ (see section 1).

Table 2: Bias of the estimates

	gamma	delta	rho	nu	sigma
Kmenta	-0.06507	-0.00042	0.00312	0.02987	0.04740
nls	0.00140	0.00113	0.00569	0.00093	0.02469
LM	0.00140	0.00113	0.00569	0.00093	0.02468
CG	0.00140	0.00113	0.00569	0.00093	0.02468
Newton	0.00140	0.00113	0.00569	0.00093	0.02468
BFGS	0.00139	0.00113	0.00568	0.00094	0.02469
Nelder-Mead	0.00139	0.00113	0.00565	0.00094	0.02470
SANN	0.00246	0.00117	0.00703	0.00063	0.02462
DE	0.00140	0.00113	0.00569	0.00093	0.02468
L-BFGS-B	0.00140	0.00113	0.00569	0.00093	0.02468
PORT	0.00140	0.00113	0.00569	0.00093	0.02468

Note: the biases are calculated based on all replications, i.e. including replications, where the algorithm warned about non-convergence; the biases calculated only with the replications, where all estimation methods converged, are mostly rather similar to the reported biases but the biases of ρ are about 3 times larger than the reported biases. The column "sigma" represents the biases of the elasticity of substitution.

other algorithms for non-linear least-squares but these differences are so small that they are negligible in practical work. The estimates of the Kmenta approximation are less precise than the estimates from the non-linear least-squares estimations.

Table 3: Root mean square error of the estimates

	gamma	delta	rho	nu	sigma
Kmenta	0.15584	0.02825	0.33360	0.07037	0.04774
nls	0.09131	0.02228	0.25347	0.03647	0.02393
LM	0.09131	0.02228	0.25346	0.03647	0.02393
CG	0.09131	0.02228	0.25346	0.03647	0.02393
Newton	0.09131	0.02228	0.25347	0.03647	0.02393
BFGS	0.09131	0.02228	0.25347	0.03647	0.02393
Nelder-Mead	0.09130	0.02228	0.25344	0.03647	0.02393
SANN	0.09324	0.02242	0.25645	0.03724	0.02463
DE	0.09131	0.02228	0.25347	0.03647	0.02393
L-BFGS-B	0.09131	0.02228	0.25347	0.03647	0.02393
PORT	0.09131	0.02228	0.25347	0.03647	0.02393

Note: the root mean square errors are calculated based on all replications, i.e. including replications, where the algorithm warned about non-convergence; the root mean square errors calculated only with the replications, where all estimation methods converged, are very close to the reported root mean square errors.

5. Conclusion

We have demonstrated several approaches to estimate the CES function, e.g. the Kmenta approximation, the Levenberg-Marquardt algorithm, several other gradient-based and global

optimisation algorithms, a grid search, and the standard tool for non-linear least-squares estimations in R, `nls`. We compared the performance of these methods in a Monte Carlo simulation. For the given data generating process, all methods proved satisfying results. Anyway, our simulation confirms other simulation studies (e.g. Thursby 1980) in respect to the unsatisfying result for the estimate of ρ. However, our results show that the elasticity of substitution σ—which is generally of interest—is close to the §true§ value. Hence, one should not range this problem as too severe.

The results were derived under the ideal lab-conditions of a simulation. It is clear that not all methods will return such satisfying results if they face real-world data. Given the econometric problems that are often caused by real-world data, the presented methods will more clearly display their fortitudes and weaknesses in empirical applications.

However, the **micEconCES** package provides the user with a multitude of instruments to address common econometric problems in estimating the CES function with real-world data. So the user should be able to find a satisfying solution for estimating the CES function in most cases.

A. Derivations of Taylor series approximations

The derivation of the Taylor series (Kmenta) approximation of the CES function in section A.1 is based on Uebe (2000). The derivation of the Talor series approximation of the partial derivatives of the CES function with respect to the coefficients in section A.2 is novel but inspired by Uebe (2000).

A.1. CES function (Kmenta approximation)

$$y = \gamma \left(\delta \, x_1^{-\rho} + (1 - \delta) \, x_2^{-\rho} \right)^{-\frac{\nu}{\rho}} \tag{30}$$

Logarithmized CES function:

$$\ln y = \ln \gamma - \frac{\nu}{\rho} \ln \left(\delta \, x_1^{-\rho} + (1 - \delta) \, x_2^{-\rho} \right) \tag{31}$$

Define function

$$f(\rho) \equiv -\frac{\nu}{\rho} \ln \left(\delta \, x_1^{-\rho} + (1 - \delta) \, x_2^{-\rho} \right) \tag{32}$$

so that

$$\ln y = \ln \gamma + f(\rho) . \tag{33}$$

Now we can approximate the logarithmized CES by a first-order Taylor series approximation around $\rho = 0$:

$$\ln y \approx \ln \gamma + f(0) + \rho f'(0) \tag{34}$$

We define function

$$g(\rho) \equiv \delta \, x_1^{-\rho} + (1 - \delta) \, x_2^{-\rho} \tag{35}$$

so that

$$f(\rho) = -\frac{\nu}{\rho} \ln (g(\rho)) . \tag{36}$$

Now we can calculate the first partial derivative of $f(\rho)$:

$$f'(\rho) = \frac{\nu}{\rho^2} \ln (g(\rho)) - \frac{\nu}{\rho} \frac{g'(\rho)}{g(\rho)} \tag{37}$$

and the first three derivatives of $g(\rho)$

$$\begin{aligned} g'(\rho) &= -\delta\, x_1^{-\rho} \ln x_1 - (1-\delta)\, x_2^{-\rho} \ln x_2 && (38)\\ g''(\rho) &= \delta\, x_1^{-\rho} (\ln x_1)^2 + (1-\delta)\, x_2^{-\rho} (\ln x_2)^2 && (39)\\ g'''(\rho) &= -\delta\, x_1^{-\rho} (\ln x_1)^3 - (1-\delta)\, x_2^{-\rho} (\ln x_2)^3 . && (40) \end{aligned}$$

At the point of approximation $\rho = 0$ we have

$$\begin{aligned} g(0) &= 1 && (41)\\ g'(0) &= -\delta \ln x_1 - (1-\delta) \ln x_2 && (42)\\ g''(0) &= \delta (\ln x_1)^2 + (1-\delta) (\ln x_2)^2 && (43)\\ g'''(0) &= -\delta (\ln x_1)^3 - (1-\delta) (\ln x_2)^3 && (44) \end{aligned}$$

Now we calculate the limit of $f(\rho)$ for $\rho \to 0$:

$$\begin{aligned} f(0) &= \lim_{\rho\to 0} f(\rho) && (45)\\ &= \lim_{\rho\to 0} \frac{-\nu \ln(g(\rho))}{\rho} && (46)\\ &= \lim_{\rho\to 0} \frac{-\nu \frac{g'(\rho)}{g(\rho)}}{1} && (47)\\ &= \nu(\delta \ln x_1 + (1-\delta) \ln x_2) && (48) \end{aligned}$$

and the limit of $f'(\rho)$ for $\rho \to 0$:

$$\begin{aligned} f'(0) &= \lim_{\rho\to 0} f'(\rho) && (49)\\ &= \lim_{\rho\to 0} \left(\frac{\nu}{\rho^2} \ln(g(\rho)) - \frac{\nu}{\rho} \frac{g'(\rho)}{g(\rho)} \right) && (50)\\ &= \lim_{\rho\to 0} \frac{\nu \ln(g(\rho)) - \nu\, \rho \frac{g'(\rho)}{g(\rho)}}{\rho^2} && (51)\\ &= \lim_{\rho\to 0} \frac{\nu \frac{g'(\rho)}{g(\rho)} - \nu \frac{g'(\rho)}{g(\rho)} - \nu\, \rho \frac{g''(\rho) g(\rho) - (g'(\rho))^2}{(g(\rho))^2}}{2\rho} && (52)\\ &= \lim_{\rho\to 0} -\frac{\nu}{2} \frac{g''(\rho)\, g(\rho) - (g'(\rho))^2}{(g(\rho))^2} && (53)\\ &= -\frac{\nu}{2} \frac{g''(0)\, g(0) - (g'(0))^2}{(g(0))^2} && (54)\\ &= -\frac{\nu}{2} \left(\delta (\ln x_1)^2 + (1-\delta) (\ln x_2)^2 - (-\delta \ln x_1 - (1-\delta) \ln x_2)^2 \right) && (55)\\ &= -\frac{\nu}{2} \Big(\delta (\ln x_1)^2 + (1-\delta) (\ln x_2)^2 - \delta^2 (\ln x_1)^2 && (56) \end{aligned}$$

$$
\begin{aligned}
& -2\delta\,(1-\delta)\,\ln x_1 \ln x_2 - (1-\delta)^2\,(\ln x_2)^2\Big) \\
= & -\frac{\nu}{2}\Big(\left(\delta-\delta^2\right)(\ln x_1)^2 + \left((1-\delta)-(1-\delta)^2\right)(\ln x_2)^2 \qquad (57)\\
& -2\delta\,(1-\delta)\,\ln x_1 \ln x_2\Big) \\
= & -\frac{\nu}{2}\Big(\delta\,(1-\delta)\,(\ln x_1)^2 + (1-\delta)\,(1-(1-\delta))\,(\ln x_2)^2 \qquad (58)\\
& -2\delta\,(1-\delta)\,\ln x_1 \ln x_2\Big) \\
= & -\frac{\nu\delta\,(1-\delta)}{2}\left((\ln x_1)^2 - 2\,\ln x_1 \ln x_2 + (\ln x_2)^2\right) \qquad (59)\\
= & -\frac{\nu\delta\,(1-\delta)}{2}\,(\ln x_1 - \ln x_2)^2 \qquad (60)
\end{aligned}
$$

so that we get following first-order Taylor series approximation around $\rho = 0$:

$$
\ln y \approx \ln\gamma + \nu\delta\,\ln x_1 + \nu\,(1-\delta)\,\ln x_2 - \nu\rho\,\delta\,(1-\delta)\,(\ln x_1 - \ln x_2)^2 \qquad (61)
$$

A.2. Derivatives with respect to coefficients

$$
\begin{aligned}
\frac{\partial y}{\partial \gamma} &= \left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} && (62)\\
\frac{\partial y}{\partial \delta} &= -\gamma\frac{\nu}{\rho}\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)^{-\frac{\nu}{\rho}-1}\left(x_1^{-\rho} - x_2^{-\rho}\right) && (63)\\
\frac{\partial y}{\partial \nu} &= -\frac{\gamma}{\rho}\,\ln\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} && (64)\\
\frac{\partial y}{\partial \rho} &= \frac{\gamma\nu}{\rho^2}\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)^{-\frac{\nu}{\rho}}\ln\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right) && (65)\\
& \quad + \frac{\gamma\nu}{\rho}\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)^{-\left(\frac{\nu}{\rho}+1\right)}\left(\delta\,x_1^{-\rho}\ln x_1 + (1-\delta)\,x_2^{-\rho}\ln x_2\right)
\end{aligned}
$$

Derivatives with respect to Gamma

$$
\begin{aligned}
\frac{\partial y}{\partial \gamma} &= \exp\left(-\frac{\nu}{\rho}\,\ln\left(\delta\,x_1^{-\rho} + (1-\delta)\,x_2^{-\rho}\right)\right) && (66)\\
&= \exp\left(f\left(\rho\right)\right) && (67)\\
&\approx \exp\left(f\left(0\right) + \rho f'\left(0\right)\right) && (68)\\
&= \exp\left(\nu\delta\,\ln x_1 + \nu\,(1-\delta)\,\ln x_2 - \frac{1}{2}\nu\rho\,\delta\,(1-\delta)\,(\ln x_1 - \ln x_2)^2\right) && (69)
\end{aligned}
$$

$$= x_1^{\nu\delta} x_2^{\nu(1-\delta)} \exp\left(-\frac{1}{2}\nu\rho\,\delta\,(1-\delta)\,(\ln x_1 - \ln x_2)^2\right) \tag{70}$$

Derivatives with respect to Delta

$$\frac{\partial y}{\partial \delta} = -\gamma\frac{\nu}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}-1}\left(x_1^{-\rho} - x_2^{-\rho}\right) \tag{71}$$

$$= -\gamma\nu\frac{x_1^{-\rho} - x_2^{-\rho}}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}-1} \tag{72}$$

Now we define the function $f_\delta(\rho)$

$$f_\delta(\rho) = \frac{x_1^{-\rho} - x_2^{-\rho}}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}-1} \tag{73}$$

$$= \frac{x_1^{-\rho} - x_2^{-\rho}}{\rho}\exp\left(-\left(\frac{\nu}{\rho}+1\right)\ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)\right) \tag{74}$$

so that we can approximate $\partial y/\partial\delta$ by using the first-order Taylor series approximation of $f_\delta(\rho)$:

$$\frac{\partial y}{\partial \delta} = -\gamma\nu\, f_\delta(\rho) \tag{75}$$

$$\approx -\gamma\nu\,\left(f_\delta(0) + \rho f'_\delta(0)\right) \tag{76}$$

Now we define the helper functions $g_\delta(\rho)$ and $h_\delta(\rho)$

$$g_\delta(\rho) = \left(\frac{\nu}{\rho}+1\right)\ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) \tag{77}$$

$$= \left(\frac{\nu}{\rho}+1\right)\ln\left(g(\rho)\right) \tag{78}$$

$$h_\delta(\rho) = \frac{x_1^{-\rho} - x_2^{-\rho}}{\rho} \tag{79}$$

with first derivatives

$$g'_\delta(\rho) = -\frac{\nu}{\rho^2}\ln\left(g(\rho)\right) + \left(\frac{\nu}{\rho}+1\right)\frac{g'(\rho)}{g(\rho)} \tag{80}$$

$$h'_\delta(\rho) = \frac{-\rho\left(\ln x_1\, x_1^{-\rho} - \ln x_2\, x_2^{-\rho}\right) - x_1^{-\rho} + x_2^{-\rho}}{\rho^2} \tag{81}$$

so that

$$f_\delta(\rho) = h_\delta(\rho)\,\exp\left(-g_\delta(\rho)\right) \tag{82}$$

and

$$f'_\delta(\rho) = h'_\delta(\rho)\exp(-g_\delta(\rho)) - h_\delta(\rho)\exp(-g_\delta(\rho))\,g'_\delta(\rho) \tag{83}$$

Now we can calculate the limits of $g_\delta(\rho)$, $g'_\delta(\rho)$, $h_\delta(\rho)$and $h'_\delta(\rho)$ for $\rho \to 0$ by

$$\begin{aligned}
g_\delta(0) &= \lim_{\rho\to 0} g_\delta(\rho) && (84)\\
&= \lim_{\rho\to 0}\left(\left(\frac{\nu}{\rho}+1\right)\ln(g(\rho))\right) && (85)\\
&= \lim_{\rho\to 0}\frac{(\nu+\rho)\ln(g(\rho))}{\rho} && (86)\\
&= \lim_{\rho\to 0}\frac{\ln(g(\rho)) + (\nu+\rho)\,\frac{g'(\rho)}{g(\rho)}}{1} && (87)\\
&= \ln(g(0)) + \nu\frac{g'(0)}{g(0)} && (88)\\
&= -\nu\delta\,\ln x_1 - \nu(1-\delta)\,\ln x_2 && (89)
\end{aligned}$$

$$\begin{aligned}
g'_\delta(0) &= \lim_{\rho\to 0}\frac{-\nu\ln(g(\rho)) + \rho(\nu+\rho)\,\frac{g'(\rho)}{g(\rho)}}{\rho^2} && (90)\\
&= \lim_{\rho\to 0}\frac{-\nu\frac{g'(\rho)}{g(\rho)} + (\nu+\rho)\,\frac{g'(\rho)}{g(\rho)} + \rho\frac{g'(\rho)}{g(\rho)} + \rho(\nu+\rho)\,\frac{g''(\rho)g(\rho)-(g'(\rho))^2}{(g(\rho))^2}}{2\rho} && (91)\\
&= \lim_{\rho\to 0}\left(\frac{-\nu\frac{g'(\rho)}{g(\rho)} + \nu\frac{g'(\rho)}{g(\rho)} + \rho\frac{g'(\rho)}{g(\rho)} + \rho\frac{g'(\rho)}{g(\rho)} + \rho(\nu+\rho)\,\frac{g''(\rho)g(\rho)-(g'(\rho))^2}{(g(\rho))^2}}{2\rho}\right) && (92)\\
&= \lim_{\rho\to 0}\left(\frac{g'(\rho)}{g(\rho)} + \frac{1}{2}(\nu+\rho)\,\frac{g''(\rho)\,g(\rho) - (g'(\rho))^2}{(g(\rho))^2}\right) && (93)\\
&= \frac{g'(0)}{g(0)} + \frac{1}{2}\nu\frac{g''(0)\,g(0) - (g'(0))^2}{(g(0))^2} && (94)\\
&= -\delta\,\ln x_1 - (1-\delta)\,\ln x_2 + \frac{\nu\delta(1-\delta)}{2}(\ln x_1 - \ln x_2)^2 && (95)
\end{aligned}$$

$$\begin{aligned}
h_\delta(0) &= \lim_{\rho\to 0}\frac{x_1^{-\rho} - x_2^{-\rho}}{\rho} && (96)\\
&= \lim_{\rho\to 0}\frac{-\ln x_1\,x_1^{-\rho} + \ln x_2\,x_2^{-\rho}}{1} && (97)\\
&= -\ln x_1 + \ln x_2 && (98)
\end{aligned}$$

$$
\begin{aligned}
h'_\delta(0) &= \lim_{\rho\to 0} \frac{-\rho\left(\ln x_1\, x_1^{-\rho} - \ln x_2\, x_2^{-\rho}\right) - x_1^{-\rho} + x_2^{-\rho}}{\rho^2} && (99)\\
&= \lim_{\rho\to 0}\left(\frac{-\left(\ln x_1\, x_1^{-\rho} - \ln x_2\, x_2^{-\rho}\right) + \rho\left(\left(\ln x_1\right)^2 x_1^{-\rho} - \left(\ln x_2\right)^2 x_2^{-\rho}\right)}{2\rho}\right. && (100)\\
&\quad \left. + \frac{\ln x_1\, x_1^{-\rho} - \ln x_2\, x_2^{-\rho}}{2\rho}\right)\\
&= \lim_{\rho\to 0} \frac{1}{2}\left(\left(\ln x_1\right)^2 x_1^{-\rho} - \left(\ln x_2\right)^2 x_2^{-\rho}\right) && (101)\\
&= \frac{1}{2}\left(\left(\ln x_1\right)^2 - \left(\ln x_2\right)^2\right) && (102)
\end{aligned}
$$

so that we can calculate the limit of $f_\delta(\rho)$and $f'_\delta(\rho)$ for $\rho \to 0$ by

$$
\begin{aligned}
f_\delta(0) &= \lim_{\rho\to 0} f_\delta(\rho) && (103)\\
&= \lim_{\rho\to 0} \left(h_\delta(\rho)\, \exp\left(-g_\delta(\rho)\right)\right) && (104)\\
&= \lim_{\rho\to 0} h_\delta(\rho)\, \lim_{\rho\to 0} \exp\left(-g_\delta(\rho)\right) && (105)\\
&= \lim_{\rho\to 0} h_\delta(\rho)\, \exp\left(-\lim_{\rho\to 0} g_\delta(\rho)\right) && (106)\\
&= h_\delta(0)\, \exp\left(-g_\delta(0)\right) && (107)\\
&= \left(-\ln x_1 + \ln x_2\right) \exp\left(\nu\delta\, \ln x_1 + \nu\left(1-\delta\right) \ln x_2\right) && (108)\\
&= \left(-\ln x_1 + \ln x_2\right) x_1^{\nu\delta} x_2^{\nu(1-\delta)} && (109)
\end{aligned}
$$

$$
\begin{aligned}
f'_\delta(0) &= \lim_{\rho\to 0} f'_\delta(\rho) && (110)\\
&= \lim_{\rho\to 0} \left(h'_\delta(\rho)\, \exp\left(-g_\delta(\rho)\right) - h_\delta(\rho)\, \exp\left(-g_\delta(\rho)\right) g'_\delta(\rho)\right) && (111)\\
&= \lim_{\rho\to 0} h'_\delta(\rho)\, \lim_{\rho\to 0} \exp\left(-g_\delta(\rho)\right) && (112)\\
&\quad - \lim_{\rho\to 0} h_\delta(\rho)\, \lim_{\rho\to 0} \exp\left(-g_\delta(\rho)\right) \lim_{\rho\to 0} g'_\delta(\rho)\\
&= \lim_{\rho\to 0} h'_\delta(\rho)\, \exp\left(-\lim_{\rho\to 0} g_\delta(\rho)\right) && (113)\\
&\quad - \lim_{\rho\to 0} h_\delta(\rho)\, \exp\left(-\lim_{\rho\to 0} g_\delta(\rho)\right) \lim_{\rho\to 0} g'_\delta(\rho)\\
&= h'_\delta(0)\, \exp\left(-g_\delta(0)\right) - h_\delta(0)\, \exp\left(-g_\delta(0)\right) g'_\delta(0) && (114)\\
&= \exp\left(-g_\delta(0)\right)\, \left(h'_\delta(0) - h_\delta(0)\, g'_\delta(0)\right) && (115)\\
&= \exp\left(\nu\delta\, \ln x_1 + \nu\left(1-\delta\right) \ln x_2\right) \left(\frac{1}{2}\left(\left(\ln x_1\right)^2 - \left(\ln x_2\right)^2\right)\right. && (116)\\
&\quad - \left(-\ln x_1 + \ln x_2\right)\\
&\quad \left.\left(-\delta\, \ln x_1 - \left(1-\delta\right) \ln x_2 + \frac{\nu\delta\left(1-\delta\right)}{2} \left(\ln x_1 - \ln x_2\right)^2\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
&= x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\frac{1}{2} \left(\ln x_1 \right)^2 - \frac{1}{2} \left(\ln x_2 \right)^2 \right. && (117)\\
&\quad + \left(\ln x_1 - \ln x_2 \right) \\
&\quad \left. \left(-\delta\, \ln x_1 - (1-\delta)\, \ln x_2 + \frac{\nu\delta\,(1-\delta)}{2} \left(\ln x_1 - \ln x_2 \right)^2 \right) \right) \\
&= x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\frac{1}{2} \left(\ln x_1 \right)^2 - \frac{1}{2} \left(\ln x_2 \right)^2 - \delta\, \left(\ln x_1 \right)^2 \right. && (118)\\
&\quad - (1-\delta) \ln x_1\, \ln x_2 + \frac{\nu\delta\,(1-\delta)}{2} \ln x_1 \left(\ln x_1 - \ln x_2 \right)^2 \\
&\quad \left. +\delta\, \ln x_1\, \ln x_2 + (1-\delta) \left(\ln x_2 \right)^2 - \frac{\nu\delta\,(1-\delta)}{2} \ln x_2 \left(\ln x_1 - \ln x_2 \right)^2 \right) \\
&= x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\left(\frac{1}{2} - \delta \right) \left(\ln x_1 \right)^2 + \left(\frac{1}{2} - \delta \right) \left(\ln x_2 \right)^2 \right. && (119)\\
&\quad \left. -2 \left(\frac{1}{2} - \delta \right) \ln x_1\, \ln x_2 + \frac{\nu\delta\,(1-\delta)}{2} \left(\ln x_1 - \ln x_2 \right) \left(\ln x_1 - \ln x_2 \right)^2 \right) \\
&= x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\left(\frac{1}{2} - \delta \right) \left(\ln x_1 - \ln x_2 \right)^2 \right. && (120)\\
&\quad \left. + \frac{\nu\delta\,(1-\delta)}{2} \left(\ln x_1 - \ln x_2 \right) \left(\ln x_1 - \ln x_2 \right)^2 \right) \\
&= x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\frac{1}{2} - \delta + \frac{\nu\delta\,(1-\delta)}{2} \left(\ln x_1 - \ln x_2 \right) \right) \left(\ln x_1 - \ln x_2 \right)^2 && (121)\\
&= \frac{1 - 2\delta + \nu\delta\,(1-\delta) \left(\ln x_1 - \ln x_2 \right)}{2}\, x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\ln x_1 - \ln x_2 \right)^2 && (122)
\end{aligned}
$$

and approximate $\partial y / \partial \delta$ by

$$
\begin{aligned}
\frac{\partial y}{\partial \delta} &\approx -\gamma\nu\, \left(f_\delta\left(0\right) + \rho f'_\delta\left(0\right) \right) && (123)\\
&= -\gamma\nu\, \left(\left(-\ln x_1 + \ln x_2 \right) x_1^{\nu\delta} x_2^{\nu(1-\delta)} \right. && (124)\\
&\quad \left. + \rho \frac{1 - 2\delta + \nu\delta\,(1-\delta) \left(\ln x_1 - \ln x_2 \right)}{2}\, x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\ln x_1 - \ln x_2 \right)^2 \right) \\
&= \gamma\nu\, \left(\left(\ln x_1 - \ln x_2 \right) x_1^{\nu\delta} x_2^{\nu(1-\delta)} \right. && (125)\\
&\quad \left. - \rho \frac{1 - 2\delta + \nu\delta\,(1-\delta) \left(\ln x_1 - \ln x_2 \right)}{2}\, x_1^{\nu\delta}\, x_2^{\nu(1-\delta)} \left(\ln x_1 - \ln x_2 \right)^2 \right) \\
&= \gamma\nu\, \left(\ln x_1 - \ln x_2 \right) x_1^{\nu\delta} x_2^{\nu(1-\delta)} && (126)\\
&\quad \left(1 - \rho \frac{1 - 2\delta + \nu\delta\,(1-\delta) \left(\ln x_1 - \ln x_2 \right)}{2}\, \left(\ln x_1 - \ln x_2 \right) \right)
\end{aligned}
$$

Derivatives with respect to Nu

$$
\frac{\partial y}{\partial \nu} = -\frac{\gamma}{\rho}\, \ln \left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho} \right) \left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho} \right)^{-\frac{\nu}{\rho}} \qquad (127)
$$

Now we define the function $f_\nu(\rho)$

$$\begin{aligned} f_\nu(\rho) &= \frac{1}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) \left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} && (128)\\ &= \frac{1}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) \exp\left(-\frac{\nu}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)\right) && (129)\end{aligned}$$

so that we can approximate $\partial y/\partial \nu$ by using the first-order Taylor series approximation of $f_\nu(\rho)$:

$$\begin{aligned} \frac{\partial y}{\partial \nu} &= -\gamma\, f_\nu(\rho) && (130)\\ &\approx -\gamma\, \left(f_\nu(0) + \rho f_\nu'(0)\right) && (131)\end{aligned}$$

Now we define the helper function $g_\nu(\rho)$

$$\begin{aligned} g_\nu(\rho) &= \frac{1}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) && (132)\\ &= \frac{1}{\rho} \ln\left(g(\rho)\right) && (133)\end{aligned}$$

with first and second derivative

$$\begin{aligned} g_\nu'(\rho) &= \frac{\rho \frac{g'(\rho)}{g(\rho)} - \ln\left(g(\rho)\right)}{\rho^2} && (134)\\ &= \frac{1}{\rho} \frac{g'(\rho)}{g(\rho)} - \frac{\ln\left(g(\rho)\right)}{\rho^2} && (135)\\ g_\nu''(\rho) &= -\frac{1}{\rho^2} \frac{g'(\rho)}{g(\rho)} + \frac{1}{\rho} \frac{g''(\rho)}{g(\rho)} - \frac{1}{\rho} \frac{\left(g'(\rho)\right)^2}{\left(g(\rho)\right)^2} + 2\frac{\ln\left(g(\rho)\right)}{\rho^3} - \frac{1}{\rho^2} \frac{g'(\rho)}{g(\rho)} && (136)\\ &= \frac{-2\rho \frac{g'(\rho)}{g(\rho)} + \rho^2 \frac{g''(\rho)}{g(\rho)} - \rho^2 \frac{\left(g'(\rho)\right)^2}{\left(g(\rho)\right)^2} + 2\ln\left(g(\rho)\right)}{\rho^3} && (137)\end{aligned}$$

and use the function $f(\rho)$ defined above so that

$$f_\nu(\rho) = g_\nu(\rho)\, \exp\left(f(\rho)\right) \tag{138}$$

and

$$f_\nu'(\rho) = g_\nu'(\rho)\, \exp\left(f(\rho)\right) + g_\nu(\rho)\, \exp\left(f(\rho)\right)\, f'(\rho) \tag{139}$$

Now we can calculate the limits of $g_\nu(\rho)$, $g_\nu'(\rho)$, and $g_\nu''(\rho)$ for $\rho \to 0$ by

$$g_\nu(0) = \lim_{\rho \to 0} g_\nu(\rho) \tag{140}$$

$$
\begin{aligned}
&= \lim_{\rho \to 0} \frac{\ln\left(g\left(\rho\right)\right)}{\rho} && (141)\\
&= \lim_{\rho \to 0} \frac{\frac{g'(\rho)}{g(\rho)}}{1} && (142)\\
&= -\delta \ln x_1 - (1-\delta) \ln x_2 && (143)
\end{aligned}
$$

$$
\begin{aligned}
g'_\nu\left(0\right) &= \lim_{\rho \to 0} g'_\nu\left(\rho\right) && (144)\\
&= \lim_{\rho \to 0} \left(\frac{1}{\rho} \frac{g'\left(\rho\right)}{g\left(\rho\right)} - \frac{\ln\left(g\left(\rho\right)\right)}{\rho^2}\right) && (145)\\
&= \lim_{\rho \to 0} \frac{\rho \frac{g'(\rho)}{g(\rho)} - \ln\left(g\left(\rho\right)\right)}{\rho^2} && (146)\\
&= \lim_{\rho \to 0} \frac{\frac{g'(\rho)}{g(\rho)} + \rho \frac{g''(\rho)g(\rho) - (g'(\rho))^2}{(g(\rho))^2} - \frac{g'(\rho)}{g(\rho)}}{2\rho} && (147)\\
&= \lim_{\rho \to 0} \frac{g''\left(\rho\right) g\left(\rho\right) - \left(g'\left(\rho\right)\right)^2}{2\left(g\left(\rho\right)\right)^2} && (148)\\
&= \frac{g''\left(0\right) g\left(0\right) - \left(g'\left(0\right)\right)^2}{2\left(g\left(0\right)\right)^2} && (149)\\
&= \frac{1}{2}\left(\delta \left(\ln x_1\right)^2 + (1-\delta)\left(\ln x_2\right)^2 - \left(-\delta \ln x_1 - (1-\delta) \ln x_2\right)^2\right) && (150)\\
&= \frac{1}{2}\Bigg(\delta \left(\ln x_1\right)^2 + (1-\delta)\left(\ln x_2\right)^2 - \delta^2 \left(\ln x_1\right)^2 && (151)\\
&\quad -2\delta (1-\delta) \ln x_1 \ln x_2 - (1-\delta)^2 \left(\ln x_2\right)^2\Bigg)\\
&= \frac{1}{2}\Bigg(\left(\delta - \delta^2\right)\left(\ln x_1\right)^2 + \left((1-\delta) - (1-\delta)^2\right)\left(\ln x_2\right)^2 && (152)\\
&\quad -2\delta (1-\delta) \ln x_1 \ln x_2\Bigg)\\
&= \frac{1}{2}\Bigg(\delta (1-\delta)\left(\ln x_1\right)^2 + (1-\delta)\left(1-(1-\delta)\right)\left(\ln x_2\right)^2 && (153)\\
&\quad -2\delta (1-\delta) \ln x_1 \ln x_2\Bigg)\\
&= \frac{\delta (1-\delta)}{2}\left(\left(\ln x_1\right)^2 - 2 \ln x_1 \ln x_2 + \left(\ln x_2\right)^2\right) && (154)\\
&= \frac{\delta (1-\delta)}{2}\left(\ln x_1 - \ln x_2\right)^2 && (155)
\end{aligned}
$$

$$
g''_\nu\left(0\right) = \lim_{\rho \to 0} g''_\nu\left(\rho\right) \quad (156)
$$

$$= \lim_{\rho \to 0} \left(\frac{-2\rho \frac{g'(\rho)}{g(\rho)} + \rho^2 \frac{g''(\rho)}{g(\rho)} - \rho^2 \frac{(g'(\rho))^2}{(g(\rho))^2} + 2\ln(g(\rho))}{\rho^3} \right) \tag{157}$$

$$= \lim_{\rho \to 0} \left(\frac{-2\frac{g'(\rho)}{g(\rho)} - 2\rho \frac{g''(\rho)}{g(\rho)} + 2\rho \frac{(g'(\rho))^2}{(g(\rho))^2} + 2\rho \frac{g''(\rho)}{g(\rho)} + \rho^2 \frac{g'''(\rho)}{g(\rho)}}{3\rho^2} \right. \tag{158}$$

$$\left. + \frac{-\rho^2 \frac{g''(\rho)g'(\rho)}{(g(\rho))^2} - 2\rho \frac{(g'(\rho))^2}{(g(\rho))^2} - 2\rho^2 \frac{g'(\rho)g''(\rho)}{(g(\rho))^2} + 2\rho^2 \frac{(g'(\rho))^3}{(g(\rho))^3} + 2\frac{g'(\rho)}{g(\rho)}}{3\rho^2} \right)$$

$$= \lim_{\rho \to 0} \left(\frac{\rho^2 \frac{g'''(\rho)}{g(\rho)} - 3\rho^2 \frac{g''(\rho)g'(\rho)}{(g(\rho))^2} + 2\rho^2 \frac{(g'(\rho))^3}{(g(\rho))^3}}{3\rho^2} \right) \tag{159}$$

$$= \lim_{\rho \to 0} \left(\frac{1}{3} \frac{g'''(\rho)}{g(\rho)} - \frac{g''(\rho) g'(\rho)}{(g(\rho))^2} + \frac{2}{3} \frac{(g'(\rho))^3}{(g(\rho))^3} \right) \tag{160}$$

$$= \frac{1}{3} \frac{g'''(0)}{g(0)} - \frac{g''(0) g'(0)}{(g(0))^2} + \frac{2}{3} \frac{(g'(0))^3}{(g(0))^3} \tag{161}$$

$$= \frac{1}{3} \left(-\delta (\ln x_1)^3 - (1-\delta)(\ln x_2)^3 \right) \tag{162}$$
$$- \left(\delta (\ln x_1)^2 + (1-\delta)(\ln x_2)^2 \right) (-\delta \ln x_1 - (1-\delta) \ln x_2)$$
$$+ \frac{2}{3} (-\delta \ln x_1 - (1-\delta) \ln x_2)^3$$

$$= -\frac{1}{3}\delta (\ln x_1)^3 - \frac{1}{3}(1-\delta)(\ln x_2)^3 + \delta^2 (\ln x_1)^3 \tag{163}$$
$$+ \delta(1-\delta)(\ln x_1)^2 \ln x_2 + \delta(1-\delta) \ln x_1 \, (\ln x_2)^2 + (1-\delta)^2 (\ln x_2)^3$$
$$+ \frac{2}{3} \left(\delta^2 (\ln x_1)^2 + 2\delta(1-\delta) \ln x_1 \ln x_2 + (1-\delta)^2 (\ln x_2)^2 \right)$$
$$(-\delta \ln x_1 - (1-\delta) \ln x_2)$$

$$= \left(\delta^2 - \frac{1}{3}\delta \right) (\ln x_1)^3 + \delta(1-\delta)(\ln x_1)^2 \ln x_2 \tag{164}$$
$$+ \delta(1-\delta) \ln x_1 \, (\ln x_2)^2 + \left((1-\delta)^2 - \frac{1}{3}(1-\delta) \right) (\ln x_2)^3$$
$$+ \frac{2}{3} \left(\delta^2 (\ln x_1)^2 + 2\delta(1-\delta) \ln x_1 \ln x_2 + (1-\delta)^2 (\ln x_2)^2 \right)$$
$$(-\delta \ln x_1 - (1-\delta) \ln x_2)$$

$$= \left(\delta^2 - \frac{1}{3}\delta \right) (\ln x_1)^3 + \delta(1-\delta)(\ln x_1)^2 \ln x_2 \tag{165}$$
$$+ \delta(1-\delta) \ln x_1 \, (\ln x_2)^2 + \left((1-\delta)^2 - \frac{1}{3}(1-\delta) \right) (\ln x_2)^3$$
$$- \frac{2}{3} \left(\delta^3 (\ln x_1)^3 + \delta^2 (1-\delta)(\ln x_1)^2 \ln x_2 + 2\delta^2 (1-\delta)(\ln x_1)^2 \ln x_2 \right.$$
$$\left. + 2\delta(1-\delta)^2 \ln x_1 (\ln x_2)^2 + \delta(1-\delta)^2 \ln x_1 (\ln x_2)^2 + (1-\delta)^3 (\ln x_2)^3 \right)$$

$$= \left(\delta^2 - \frac{1}{3}\delta \right) (\ln x_1)^3 + \delta(1-\delta)(\ln x_1)^2 \ln x_2 \tag{166}$$

$$
\begin{aligned}
& +\delta\left(1-\delta\right)\ln x_1\ \left(\ln x_2\right)^2+\left(\left(1-\delta\right)^2-\frac{1}{3}\left(1-\delta\right)\right)\left(\ln x_2\right)^3 \\
& -\frac{2}{3}\delta^3\left(\ln x_1\right)^3-2\delta^2\left(1-\delta\right)\left(\ln x_1\right)^2\ \ln x_2-2\delta\left(1-\delta\right)^2\ln x_1\left(\ln x_2\right)^2 \\
& -\frac{2}{3}\left(1-\delta\right)^3\left(\ln x_2\right)^3 \\
= & \left(\delta^2-\frac{1}{3}\delta-\frac{2}{3}\delta^3\right)\left(\ln x_1\right)^3+\left(\delta\left(1-\delta\right)-2\delta^2\left(1-\delta\right)\right)\left(\ln x_1\right)^2\ln x_2 \qquad (167) \\
& +\left(\delta\left(1-\delta-2\delta\left(1-\delta\right)^2\right)\right)\ln x_1\ \left(\ln x_2\right)^2 \\
& +\left(\left(1-\delta\right)^2-\frac{1}{3}\left(1-\delta\right)-\frac{2}{3}\left(1-\delta\right)^3\right)\left(\ln x_2\right)^3 \\
= & \left(\delta-\frac{1}{3}-\frac{2}{3}\delta^2\right)\delta\left(\ln x_1\right)^3+\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2 \qquad (168) \\
& +\left(1-2\left(1-\delta\right)\right)\delta\left(1-\delta\right)\ln x_1\ \left(\ln x_2\right)^2 \\
& +\left(\left(1-\delta\right)-\frac{1}{3}-\frac{2}{3}\left(1-\delta\right)^2\right)\left(1-\delta\right)\left(\ln x_2\right)^3 \\
= & \left(-\frac{1}{3}+\frac{2}{3}\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^3+\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2 \qquad (169) \\
& +\left(2\delta-1\right)\delta\left(1-\delta\right)\ln x_1\ \left(\ln x_2\right)^2 \\
& +\left(1-\delta-\frac{1}{3}-\frac{2}{3}+\frac{4}{3}\delta-\frac{2}{3}\delta^2\right)\left(1-\delta\right)\left(\ln x_2\right)^3 \\
= & \left(-\frac{1}{3}+\frac{2}{3}\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^3+\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2 \qquad (170) \\
& +\left(2\delta-1\right)\delta\left(1-\delta\right)\ln x_1\ \left(\ln x_2\right)^2+\left(\frac{1}{3}-\frac{2}{3}\delta\right)\delta\left(1-\delta\right)\left(\ln x_2\right)^3 \\
= & -\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^3+\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2 \qquad (171) \\
& -\left(1-2\delta\right)\delta\left(1-\delta\right)\ln x_1\ \left(\ln x_2\right)^2+\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_2\right)^3 \\
= & -\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right) \qquad (172) \\
& \left(\left(\ln x_1\right)^3+3\left(\ln x_1\right)^2\ln x_2+3\ln x_1\ \left(\ln x_2\right)^2-\left(\ln x_2\right)^3\right) \\
= & -\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1-\ln x_2\right)^3 \qquad (173)
\end{aligned}
$$

so that we can calculate the limit of $f_\nu\left(\rho\right)$and $f'_\nu\left(\rho\right)$ for $\rho\to 0$ by

$$
\begin{aligned}
f_\nu\left(0\right) &= \lim_{\rho\to 0} f_\nu\left(\rho\right) && (174)\\
&= \lim_{\rho\to 0}\left(g_\nu\left(\rho\right)\ \exp\left(f\left(\rho\right)\right)\right) && (175)\\
&= \lim_{\rho\to 0} g_\nu\left(\rho\right)\ \lim_{\rho\to 0}\exp\left(f\left(\rho\right)\right) && (176)\\
&= \lim_{\rho\to 0} g_\nu\left(\rho\right)\ \exp\left(\lim_{\rho\to 0} f\left(\rho\right)\right) && (177)\\
&= g_\nu\left(0\right)\ \exp\left(f\left(0\right)\right) && (178)
\end{aligned}
$$

$$= (-\delta \ln x_1 - (1-\delta) \ln x_2) \exp(\nu(\delta \ln x_1 + (1-\delta) \ln x_2)) \quad (179)$$

$$= -(\delta \ln x_1 + (1-\delta) \ln x_2) x_1^{\nu\delta} x_2^{\nu(1-\delta)} \quad (180)$$

$$\begin{aligned}
f'_\nu(0) &= \lim_{\rho\to 0} f'_\nu(\rho) && (181)\\
&= \lim_{\rho\to 0} \left(g'_\nu(\rho) \exp(f(\rho)) + g_\nu(\rho) \exp(f(\rho))\ f'(\rho)\right) && (182)\\
&= \lim_{\rho\to 0} g'_\nu(\rho) \exp\left(\lim_{\rho\to 0} f(\rho)\right) + \lim_{\rho\to 0} g_\nu(\rho) \exp\left(\lim_{\rho\to 0} f(\rho)\right) \lim_{\rho\to 0} f'(\rho) && (183)\\
&= g'_\nu(0) \exp(f(0)) + g_\nu(0) \exp(f(0)) f'(0) && (184)\\
&= \exp(f(0)) \left(g'_\nu(0) + g_\nu(0)\ f'(0)\right) && (185)\\
&= \exp(\nu(\delta \ln x_1 + (1-\delta) \ln x_2)) \left(\frac{\delta(1-\delta)}{2} (\ln x_1 - \ln x_2)^2 \right. && (186)\\
&\quad \left. + (-\delta \ln x_1 - (1-\delta) \ln x_2) \left(-\frac{\nu\delta(1-\delta)}{2} (\ln x_1 - \ln x_2)^2\right)\right)\\
&= x_1^{\nu\delta} x_2^{\nu(1-\delta)} \frac{\delta(1-\delta)}{2} (\ln x_1 - \ln x_2)^2 (1 + \nu(\delta \ln x_1 + (1-\delta) \ln x_2)) && (187)
\end{aligned}$$

and approximate $\partial y / \partial \nu$ by

$$\begin{aligned}
\frac{\partial y}{\partial \nu} &\approx -\gamma\ (f_\nu(0) + \rho f'_\nu(0)) && (188)\\
&= \gamma(\delta \ln x_1 + (1-\delta) \ln x_2) x_1^{\nu\delta} x_2^{\nu(1-\delta)} && (189)\\
&\quad -\gamma\rho x_1^{\nu\delta} x_2^{\nu(1-\delta)} \frac{\delta(1-\delta)}{2} (\ln x_1 - \ln x_2)^2\\
&\quad (1 + \nu(\delta \ln x_1 + (1-\delta) \ln x_2))\\
&= \gamma x_1^{\nu\delta} x_2^{\nu(1-\delta)} \left(\delta \ln x_1 + (1-\delta) \ln x_2\right. && (190)\\
&\quad \left. -\frac{\rho\delta(1-\delta)}{2} (\ln x_1 - \ln x_2)^2 (1 + \nu(\delta \ln x_1 + (1-\delta) \ln x_2))\right)
\end{aligned}$$

Derivatives with respect to Rho

$$\begin{aligned}
\frac{\partial y}{\partial \rho} &= \frac{\gamma\nu}{\rho^2} \left(\delta x_1^{-\rho} + (1-\delta)\ x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \ln\left(\delta x_1^{-\rho} + (1-\delta)\ x_2^{-\rho}\right) && (191)\\
&\quad + \frac{\gamma\nu}{\rho} \left(\delta x_1^{-\rho} + (1-\delta)\ x_2^{-\rho}\right)^{-\left(\frac{\nu}{\rho}+1\right)} \left(\delta x_1^{-\rho} \ln x_1 + (1-\delta)\ x_2^{-\rho} \ln x_2\right)\\
&= \gamma\nu \left(\frac{1}{\rho^2} \left(\delta x_1^{-\rho} + (1-\delta)\ x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \ln\left(\delta x_1^{-\rho} + (1-\delta)\ x_2^{-\rho}\right)\right. && (192)\\
&\quad \left. + \frac{1}{\rho} \left(\delta x_1^{-\rho} + (1-\delta)\ x_2^{-\rho}\right)^{-\left(\frac{\nu}{\rho}+1\right)} \left(\delta x_1^{-\rho} \ln x_1 + (1-\delta)\ x_2^{-\rho} \ln x_2\right)\right)
\end{aligned}$$

Now we define the function $f_\rho(\rho)$

$$\begin{aligned} f_\rho(\rho) &= \frac{1}{\rho^2}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) \\ &\quad + \frac{1}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\left(\frac{\nu}{\rho}+1\right)} \left(\delta\, x_1^{-\rho} \ln x_1 + (1-\delta)\, x_2^{-\rho} \ln x_2\right) \end{aligned} \tag{193}$$

so that we can approximate $\partial y/\partial\rho$ by using the first-order Taylor series approximation of $f_\rho(\rho)$:

$$\frac{\partial y}{\partial \rho} = \gamma\nu\, f_\rho(\rho) \tag{194}$$

$$\approx \gamma\nu\, \left(f_\rho(0) + \rho f'_\rho(0)\right) \tag{195}$$

We define the helper function $g_\rho(\rho)$

$$g_\rho(\rho) = \delta\, x_1^{-\rho} \ln x_1 + (1-\delta)\, x_2^{-\rho} \ln x_2 \tag{196}$$

with first and second derivative

$$g'_\rho(\rho) = -\delta\, x_1^{-\rho} (\ln x_1)^2 - (1-\delta)\, x_2^{-\rho} (\ln x_2)^2 \tag{197}$$

$$g''_\rho(\rho) = \delta\, x_1^{-\rho} (\ln x_1)^3 + (1-\delta)\, x_2^{-\rho} (\ln x_2)^3 \tag{198}$$

and use the functions $g(\rho)$ and $g_\nu(\rho)$ all defined above so that

$$\begin{aligned} f_\rho(\rho) &= \frac{1}{\rho^2}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) \\ &\quad + \frac{1}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\left(\frac{\nu}{\rho}+1\right)} \left(\delta\, x_1^{-\rho} \ln x_1 + (1-\delta)\, x_2^{-\rho} \ln x_2\right) \end{aligned} \tag{199}$$

$$\begin{aligned} &= \frac{1}{\rho^2}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right) \\ &\quad + \frac{1}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-1} \\ &\quad \left(\delta\, x_1^{-\rho} \ln x_1 + (1-\delta)\, x_2^{-\rho} \ln x_2\right) \end{aligned} \tag{200}$$

$$\begin{aligned} &= \frac{1}{\rho}\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-\frac{\nu}{\rho}} \left(\frac{1}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)\right. \\ &\quad \left. + \left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-1} \left(\delta\, x_1^{-\rho} \ln x_1 + (1-\delta)\, x_2^{-\rho} \ln x_2\right)\right) \end{aligned} \tag{201}$$

$$\begin{aligned} &= \frac{1}{\rho} \exp\left(-\frac{\nu}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)\right) \left(\frac{1}{\rho} \ln\left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)\right. \\ &\quad \left. + \left(\delta\, x_1^{-\rho} + (1-\delta)\, x_2^{-\rho}\right)^{-1} \left(\delta\, x_1^{-\rho} \ln x_1 + (1-\delta)\, x_2^{-\rho} \ln x_2\right)\right) \end{aligned} \tag{202}$$

$$= \frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)\left(g_\nu\left(\rho\right)+g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\right)}{\rho} \tag{203}$$

and we can calculate its first derivative

$$\begin{aligned} f'_\rho\left(\rho\right) &= \frac{-\rho\nu\,\exp\left(-\nu g_\nu\left(\rho\right)\right)\,g'_\nu\left(\rho\right)\left(g_\nu\left(\rho\right)+g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\right)}{\rho^2} \qquad (204)\\ &\quad+\frac{\rho\exp\left(-\nu g_\nu\left(\rho\right)\right)\left(g'_\nu\left(\rho\right)-g\left(\rho\right)^{-2}g'\left(\rho\right)g_\rho\left(\rho\right)+g\left(\rho\right)^{-1}g'_\rho\left(\rho\right)\right)}{\rho^2}\\ &\quad-\frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)\left(g_\nu\left(\rho\right)+g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\right)}{\rho^2}\\ &= \frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)\rho\left(-\nu g'_\nu\left(\rho\right)g_\nu\left(\rho\right)-\nu g'_\nu\left(\rho\right)g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\right)}{\rho^2} \qquad (205)\\ &\quad+\frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)\rho\left(g'_\nu\left(\rho\right)-g\left(\rho\right)^{-2}g'\left(\rho\right)g_\rho\left(\rho\right)+g\left(\rho\right)^{-1}g'_\rho\left(\rho\right)\right)}{\rho^2}\\ &\quad-\frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)\left(g_\nu\left(\rho\right)+g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\right)}{\rho^2}\\ &= \frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)}{\rho^2}\Big(\rho\Big(-\nu g'_\nu\left(\rho\right)g_\nu\left(\rho\right)-\nu g'_\nu\left(\rho\right)g\left(\rho\right)^{-1}g_\rho\left(\rho\right) \qquad (206)\\ &\quad+g'_\nu\left(\rho\right)-g\left(\rho\right)^{-2}g'\left(\rho\right)g_\rho\left(\rho\right)+g\left(\rho\right)^{-1}g'_\rho\left(\rho\right)\Big)\\ &\quad-g_\nu\left(\rho\right)-g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\Big)\\ &= \frac{\exp\left(-\nu g_\nu\left(\rho\right)\right)}{\rho^2}\Big(-\nu\rho g'_\nu\left(\rho\right)g_\nu\left(\rho\right)-\nu\rho g'_\nu\left(\rho\right)g\left(\rho\right)^{-1}g_\rho\left(\rho\right) \qquad (207)\\ &\quad+\rho g'_\nu\left(\rho\right)-\rho g\left(\rho\right)^{-2}g'\left(\rho\right)g_\rho\left(\rho\right)+\rho g\left(\rho\right)^{-1}g'_\rho\left(\rho\right)\\ &\quad-g_\nu\left(\rho\right)-g\left(\rho\right)^{-1}g_\rho\left(\rho\right)\Big)\end{aligned}$$

Now we can calculate the limits of $g_\rho\left(\rho\right)$, $g'_\rho\left(\rho\right)$, and $g''_\rho\left(\rho\right)$ for $\rho\to 0$ by

$$\begin{aligned} g_\rho\left(0\right) &= \lim_{\rho\to 0} g_\rho\left(\rho\right) \qquad (208)\\ &= \lim_{\rho\to 0}\left(\delta\,x_1^{-\rho}\ln x_1+\left(1-\delta\right)\,x_2^{-\rho}\ln x_2\right) \qquad (209)\\ &= \delta\,\ln x_1\,\lim_{\rho\to 0}x_1^{-\rho}+\left(1-\delta\right)\,\ln x_2\,\lim_{\rho\to 0}x_2^{-\rho} \qquad (210)\\ &= \delta\,\ln x_1+\left(1-\delta\right)\,\ln x_2 \qquad (211)\end{aligned}$$

$$\begin{aligned} g'_\rho\left(0\right) &= \lim_{\rho\to 0} g'_\rho\left(\rho\right) \qquad (212)\\ &= \lim_{\rho\to 0}\left(-\delta\,x_1^{-\rho}\left(\ln x_1\right)^2-\left(1-\delta\right)\,x_2^{-\rho}\left(\ln x_2\right)^2\right) \qquad (213)\end{aligned}$$

$$= -\delta (\ln x_1)^2 \lim_{\rho \to 0} x_1^{-\rho} - (1-\delta) (\ln x_2)^2 \lim_{\rho \to 0} x_2^{-\rho} \tag{214}$$

$$= -\delta (\ln x_1)^2 - (1-\delta) (\ln x_2)^2 \tag{215}$$

$$g''_\rho (0) = \lim_{\rho \to 0} g''_\rho (\rho) \tag{216}$$

$$= \lim_{\rho \to 0} \left(\delta \, x_1^{-\rho} (\ln x_1)^3 + (1-\delta) \, x_2^{-\rho} (\ln x_2)^3 \right) \tag{217}$$

$$= \delta (\ln x_1)^3 \lim_{\rho \to 0} x_1^{-\rho} + (1-\delta) (\ln x_2)^3 \lim_{\rho \to 0} x_2^{-\rho} \tag{218}$$

$$= \delta (\ln x_1)^3 + (1-\delta) (\ln x_2)^3 \tag{219}$$

so that we can calculate the limit of $f_\rho (\rho)$ for $\rho \to 0$ by

$$f_\rho (0) = \lim_{\rho \to 0} f_\rho (\rho) \tag{220}$$

$$= \lim_{\rho \to 0} \left(\frac{\exp (-\nu g_\nu (\rho)) \left(g_\nu (\rho) + g (\rho)^{-1} g_\rho (\rho) \right)}{\rho} \right) \tag{221}$$

$$= \lim_{\rho \to 0} \left(\frac{-\nu \, \exp (-\nu g_\nu (\rho)) \, g'_\nu (\rho) \left(g_\nu (\rho) + g (\rho)^{-1} g_\rho (\rho) \right)}{1} \right. \tag{222}$$

$$\left. + \frac{\exp (-\nu g_\nu (\rho)) \left(g'_\nu (\rho) - g (\rho)^{-2} g' (\rho) \, g_\rho (\rho) + g (\rho)^{-1} g'_\rho (\rho) \right)}{1} \right)$$

$$= -\nu \, \exp (-\nu g_\nu (0)) \, g'_\nu (0) \left(g_\nu (0) + g (0)^{-1} g_\rho (0) \right) \tag{223}$$

$$+ \exp (-\nu g_\nu (0)) \left(g'_\nu (0) - g (0)^{-2} g' (0) \, g_\rho (0) + g (0)^{-1} g'_\rho (0) \right)$$

$$= \exp (-\nu g_\nu (0)) \left(-\nu \, g'_\nu (0) \right) \left(g_\nu (0) + g (0)^{-1} g_\rho (0) \right) \tag{224}$$

$$+ \exp (-\nu g_\nu (0)) \left(g'_\nu (0) - g (0)^{-2} g' (0) \, g_\rho (0) + g (0)^{-1} g'_\rho (0) \right)$$

$$= \exp (-\nu g_\nu (0)) \left(\left(-\nu \, g'_\nu (0) \right) \left(g_\nu (0) + g (0)^{-1} g_\rho (0) \right) \right. \tag{225}$$

$$\left. + g'_\nu (0) - g (0)^{-2} g' (0) \, g_\rho (0) + g (0)^{-1} g'_\rho (0) \right)$$

$$= \exp (-\nu (-\delta \, \ln x_1 - (1-\delta) \, \ln x_2)) \left(-\frac{\nu \delta (1-\delta)}{2} (\ln x_1 - \ln x_2)^2 \right. \tag{226}$$

$$(-\delta \, \ln x_1 - (1-\delta) \, \ln x_2 + \delta \, \ln x_1 + (1-\delta) \, \ln x_2)$$

$$+ \frac{\delta (1-\delta)}{2} (\ln x_1 - \ln x_2)^2$$

$$- (-\delta \, \ln x_1 - (1-\delta) \, \ln x_2) (\delta \, \ln x_1 + (1-\delta) \, \ln x_2)$$

$$\left. - \delta (\ln x_1)^2 - (1-\delta) (\ln x_2)^2 \right)$$

$$= x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \left(\frac{1}{2} \delta (1-\delta) (\ln x_1)^2 - \delta (1-\delta) \ln x_1 \ln x_2 \right. \tag{227}$$

$$\left. + \frac{1}{2} \delta (1-\delta) (\ln x_2)^2 + \delta^2 (\ln x_1)^2 + 2\delta (1-\delta) \ln x_1 \ln x_2 \right.$$

$$
\begin{aligned}
& + (1-\delta)^2 \, (\ln x_2)^2 - \delta \, (\ln x_1)^2 - (1-\delta) \, (\ln x_2)^2 \Big) \\
= \; & x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \left(\left(\frac{1}{2} \delta \, (1-\delta) + \delta^2 - \delta \right) (\ln x_1)^2 \right. && (228) \\
& + \left(\frac{1}{2} \delta \, (1-\delta) + (1-\delta)^2 - (1-\delta) \right) \\
& \left. (\ln x_2)^2 + \delta \, (1-\delta) \ln x_1 \ln x_2 \right) \\
= \; & x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \left(\left(\frac{1}{2} \delta - \frac{1}{2} \delta^2 + \delta^2 - \delta \right) (\ln x_1)^2 \right. && (229) \\
& + \left(\frac{1}{2} \delta - \frac{1}{2} \delta^2 + 1 - 2\delta + \delta^2 - 1 + \delta \right) (\ln x_2)^2 \\
& \left. + \delta \, (1-\delta) \ln x_1 \ln x_2 \right) \\
= \; & x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \left(\left(-\frac{1}{2} \delta + \frac{1}{2} \delta^2 \right) (\ln x_1)^2 \right. && (230) \\
& \left. + \left(-\frac{1}{2} \delta + \frac{1}{2} \delta^2 \right) (\ln x_2)^2 + \delta \, (1-\delta) \ln x_1 \ln x_2 \right) \\
= \; & x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \left(-\frac{1}{2} \delta \, (1-\delta) \, (\ln x_1)^2 \right. && (231) \\
& \left. - \frac{1}{2} \delta \, (1-\delta) \, (\ln x_2)^2 + \delta \, (1-\delta) \ln x_1 \ln x_2 \right) \\
= \; & -\frac{1}{2} \delta \, (1-\delta) \, x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \left((\ln x_1)^2 - 2 \ln x_1 \ln x_2 + (\ln x_2)^2 \right) && (232) \\
= \; & -\frac{1}{2} \delta \, (1-\delta) \, x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} \, (\ln x_1 - \ln x_2)^2 && (233)
\end{aligned}
$$

Before we can apply de l'Hospital's rule to $\lim_{\rho \to 0} f'_\rho(\rho)$, we have to check whether also the numerator converges to zero. We do this by defining a helper function $h_\rho(\rho)$, where the numerator converges to zero if $h_\rho(\rho)$ converges to zero for $\rho \to 0$

$$
\begin{aligned}
h_\rho(\rho) &= -g_\nu(\rho) - g(\rho)^{-1} \, g_\rho(\rho) && (234) \\
h_\rho(0) &= \lim_{\rho \to 0} h_\rho(\rho) && (235) \\
&= \lim_{\rho \to 0} \left(-g_\nu(\rho) - g(\rho)^{-1} \, g_\rho(\rho) \right) && (236) \\
&= -g_\nu(0) - g(0)^{-1} \, g_\rho(0) && (237) \\
&= -\left(-\delta \, \ln x_1 - (1-\delta) \, \ln x_2 \right) - \left(\delta \, \ln x_1 + (1-\delta) \, \ln x_2 \right) && (238) \\
&= 0 && (239)
\end{aligned}
$$

As both the numerator and the denominator converge to zero, we can calculate $\lim_{\rho \to 0} f'_\rho(\rho)$

by using de l'Hospital's rule.

$$
\begin{aligned}
f'_\rho(0) &= \lim_{\rho\to 0} f'_\rho(\rho) && (240)\\
&= \lim_{\rho\to 0}\Bigg(\frac{\exp\left(-\nu g_\nu(\rho)\right)}{\rho^2}\left(-\nu\rho g'_\nu(\rho) g_\nu(\rho)\right. && (241)\\
&\quad -\nu\rho g'_\nu(\rho) g(\rho)^{-1} g_\rho(\rho) + \rho g'_\nu(\rho) - \rho g(\rho)^{-2} g'(\rho) g_\rho(\rho)\\
&\quad \left.+\rho g(\rho)^{-1} g'_\rho(\rho) - g_\nu(\rho) - g(\rho)^{-1} g_\rho(\rho)\right)\Bigg)\\
&= \lim_{\rho\to 0}\left(\exp\left(-\nu g_\nu(\rho)\right)\right)\lim_{\rho\to 0}\Bigg(\frac{1}{\rho^2}\left(-\nu\rho g'_\nu(\rho) g_\nu(\rho)\right. && (242)\\
&\quad -\nu\rho g'_\nu(\rho) g(\rho)^{-1} g_\rho(\rho) + \rho g'_\nu(\rho) - \rho g(\rho)^{-2} g'(\rho) g_\rho(\rho)\\
&\quad \left.+\rho g(\rho)^{-1} g'_\rho(\rho) - g_\nu(\rho) - g(\rho)^{-1} g_\rho(\rho)\right)\Bigg)\\
&= \lim_{\rho\to 0}\left(\exp\left(-\nu g_\nu(\rho)\right)\right)\lim_{\rho\to 0}\Bigg(\frac{1}{2\rho}\left(-\nu g'_\nu(\rho) g_\nu(\rho) - \nu\rho g''_\nu(\rho) g_\nu(\rho)\right. && (243)\\
&\quad -\nu\rho g'_\nu(\rho) g'_\nu(\rho) - \nu g'_\nu(\rho) g(\rho)^{-1} g_\rho(\rho) - \nu\rho g''_\nu(\rho) g(\rho)^{-1} g_\rho(\rho)\\
&\quad +\nu\rho g'_\nu(\rho) g(\rho)^{-2} g'(\rho) g_\rho(\rho) - \nu\rho g'_\nu(\rho) g(\rho)^{-1} g'_\rho(\rho) + g'_\nu(\rho)\\
&\quad +\rho g''_\nu(\rho) - g(\rho)^{-2} g'(\rho) g_\rho(\rho) + 2\rho g(\rho)^{-3}\left(g'(\rho)\right)^2 g_\rho(\rho)\\
&\quad -\rho g(\rho)^{-2} g''(\rho) g_\rho(\rho) - \rho g(\rho)^{-2} g'(\rho) g'_\rho(\rho) + g(\rho)^{-1} g'_\rho(\rho)\\
&\quad -\rho g(\rho)^{-2} g'(\rho) g'_\rho(\rho) + \rho g(\rho)^{-1} g''_\rho(\rho)\\
&\quad \left.-g'_\nu(\rho) + g(\rho)^{-2} g'(\rho) g_\rho(\rho) - g(\rho)^{-1} g'_\rho(\rho)\right)\Bigg)\\
&= \frac{1}{2}\lim_{\rho\to 0}\left(\exp\left(-\nu g_\nu(\rho)\right)\right)\lim_{\rho\to 0}\Bigg(\frac{1}{\rho}\left(-\nu g'_\nu(\rho) g_\nu(\rho) - \nu\rho g''_\nu(\rho) g_\nu(\rho)\right. && (244)\\
&\quad -\nu\rho g'_\nu(\rho) g'_\nu(\rho) - \nu g'_\nu(\rho) g(\rho)^{-1} g_\rho(\rho) - \nu\rho g''_\nu(\rho) g(\rho)^{-1} g_\rho(\rho)\\
&\quad +\nu\rho g'_\nu(\rho) g(\rho)^{-2} g'(\rho) g_\rho(\rho) - \nu\rho g'_\nu(\rho) g(\rho)^{-1} g'_\rho(\rho) + \rho g''_\nu(\rho)\\
&\quad +2\rho g(\rho)^{-3}\left(g'(\rho)\right)^2 g_\rho(\rho) - \rho g(\rho)^{-2} g''(\rho) g_\rho(\rho)\\
&\quad \left.-2\rho g(\rho)^{-2} g'(\rho) g'_\rho(\rho) + \rho g(\rho)^{-1} g''_\rho(\rho)\right)\Bigg)\\
&= \frac{1}{2}\lim_{\rho\to 0}\left(\exp\left(-\nu g_\nu(\rho)\right)\right) && (245)\\
&\quad \lim_{\rho\to 0}\Bigg(\frac{1}{\rho}\left(-\nu g'_\nu(\rho) g_\nu(\rho) - \nu g'_\nu(\rho) g(\rho)^{-1} g_\rho(\rho)\right)\\
&\quad -\nu g''_\nu(\rho) g_\nu(\rho) - \nu g'_\nu(\rho) g'_\nu(\rho) - \nu g''_\nu(\rho) g(\rho)^{-1} g_\rho(\rho)\\
&\quad +\nu g'_\nu(\rho) g(\rho)^{-2} g'(\rho) g_\rho(\rho) - \nu g'_\nu(\rho) g(\rho)^{-1} g'_\rho(\rho)\\
&\quad +g''_\nu(\rho) + 2g(\rho)^{-3}\left(g'(\rho)\right)^2 g_\rho(\rho) - g(\rho)^{-2} g''(\rho) g_\rho(\rho)\\
&\quad \left.-2g(\rho)^{-2} g'(\rho) g'_\rho(\rho) + g(\rho)^{-1} g''_\rho(\rho)\right)\\
&= \frac{1}{2}\lim_{\rho\to 0}\left(\exp\left(-\nu g_\nu(\rho)\right)\right) && (246)
\end{aligned}
$$

$$\begin{aligned}
&\left(\lim_{\rho\to 0}\left(\frac{1}{\rho}\left(-\nu g'_\nu(\rho)\, g_\nu(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho)\right)\right)\right.\\
&+\lim_{\rho\to 0}\left(-\nu g''_\nu(\rho)\, g_\nu(\rho) - \nu g'_\nu(\rho)\, g'_\nu(\rho) - \nu g''_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho)\right.\\
&+\nu g'_\nu(\rho)\, g(\rho)^{-2} g'(\rho)\, g_\rho(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g'_\rho(\rho) + g''_\nu(\rho)\\
&+2g(\rho)^{-3}\left(g'(\rho)\right)^2 g_\rho(\rho) - g(\rho)^{-2} g''(\rho)\, g_\rho(\rho)\\
&\left.\left.-2g(\rho)^{-2} g'(\rho)\, g'_\rho(\rho) + g(\rho)^{-1} g''_\rho(\rho)\right)\right)
\end{aligned}$$

Before we can apply de l'Hospital's rule again, we have to check if also the numerator converges to zero. We do this by defining a helper function $k_\rho(\rho)$, where the numerator converges to zero if $k_\rho(\rho)$ converges to zero for $\rho \to 0$

$$\begin{aligned}
k_\rho(\rho) &= -\nu g'_\nu(\rho)\, g_\nu(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho) && (247)\\
k_\rho(0) &= \lim_{\rho\to 0} k_\rho(\rho) && (248)\\
&= \lim_{\rho\to 0}\left(-\nu g'_\nu(\rho)\, g_\nu(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho)\right) && (249)\\
&= -\nu g'_\nu(0)\, g_\nu(0) - \nu g'_\nu(0)\, g(0)^{-1} g_\rho(0) && (250)\\
&= -\frac{\nu\delta(1-\delta)}{2}\left(\ln x_1 - \ln x_2\right)^2\left(-\delta\ln x_1 - (1-\delta)\ln x_2\right) && (251)\\
&\quad -\frac{\nu\delta(1-\delta)}{2}\left(\ln x_1 - \ln x_2\right)^2\left(\delta\ln x_1 + (1-\delta)\ln x_2\right)\\
&= 0 && (252)
\end{aligned}$$

As both the numerator and the denominator converge to zero, we can apply de l'Hospital's rule.

$$\begin{aligned}
\lim_{\rho\to 0}\frac{k_\rho(\rho)}{\rho} &= \lim_{\rho\to 0} k_\rho(\rho) && (253)\\
&= \lim_{\rho\to 0}\left(-\nu g''_\nu(\rho)\, g_\nu(\rho) - \nu\left(g'_\nu(\rho)\right)^2 - \nu g''_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho)\right. && (254)\\
&\quad \left.+\nu g'_\nu(\rho)\, g(\rho)^{-2} g'(\rho)\, g_\rho(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g'_\rho(\rho)\right)
\end{aligned}$$

and hence,

$$\begin{aligned}
f'_\rho(0) &= \frac{1}{2}\lim_{\rho\to 0}\left(\exp\left(-\nu g_\nu(\rho)\right)\right) && (255)\\
&\quad \left(\lim_{\rho\to 0}\left(\frac{1}{\rho}\left(-\nu g'_\nu(\rho)\, g_\nu(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho)\right)\right)\right.\\
&\quad +\lim_{\rho\to 0}\left(-\nu g''_\nu(\rho)\, g_\nu(\rho) - \nu g'_\nu(\rho)\, g'_\nu(\rho) - \nu g''_\nu(\rho)\, g(\rho)^{-1} g_\rho(\rho)\right.\\
&\quad +\nu g'_\nu(\rho)\, g(\rho)^{-2} g'(\rho)\, g_\rho(\rho) - \nu g'_\nu(\rho)\, g(\rho)^{-1} g'_\rho(\rho)\\
&\quad \left.\left.+g''_\nu(\rho) + 2g(\rho)^{-3}\left(g'(\rho)\right)^2 g_\rho(\rho) - g(\rho)^{-2} g''(\rho)\, g_\rho(\rho)\right)\right)
\end{aligned}$$

$$
\begin{aligned}
& -2g\left(\rho\right)^{-2} g'\left(\rho\right) g'_{\rho}\left(\rho\right) + g\left(\rho\right)^{-1} g''_{\rho}\left(\rho\right)\Big)\Bigg) \\
= \; & \frac{1}{2} \lim_{\rho \to 0} \left(\exp\left(-\nu g_{\nu}\left(\rho\right)\right)\right) \qquad (256) \\
& \Bigg(\lim_{\rho \to 0} \Big(-\nu g''_{\nu}\left(\rho\right) g_{\nu}\left(\rho\right) - \nu\left(g'_{\nu}\left(\rho\right)\right)^{2} - \nu g''_{\nu}\left(\rho\right) g\left(\rho\right)^{-1} g_{\rho}\left(\rho\right) \\
& + \nu g'_{\nu}\left(\rho\right) g\left(\rho\right)^{-2} g'\left(\rho\right) g_{\rho}\left(\rho\right) - \nu g'_{\nu}\left(\rho\right) g\left(\rho\right)^{-1} g'_{\rho}\left(\rho\right)\Big) \\
& + \lim_{\rho \to 0} \Big(-\nu g''_{\nu}\left(\rho\right) g_{\nu}\left(\rho\right) - \nu g'_{\nu}\left(\rho\right) g'_{\nu}\left(\rho\right) - \nu g''_{\nu}\left(\rho\right) g\left(\rho\right)^{-1} g_{\rho}\left(\rho\right) \\
& + \nu g'_{\nu}\left(\rho\right) g\left(\rho\right)^{-2} g'\left(\rho\right) g_{\rho}\left(\rho\right) - \nu g'_{\nu}\left(\rho\right) g\left(\rho\right)^{-1} g'_{\rho}\left(\rho\right) + g''_{\nu}\left(\rho\right) \\
& + 2g\left(\rho\right)^{-3} \left(g'\left(\rho\right)\right)^{2} g_{\rho}\left(\rho\right) - g\left(\rho\right)^{-2} g''\left(\rho\right) g_{\rho}\left(\rho\right) \\
& - 2g\left(\rho\right)^{-2} g'\left(\rho\right) g'_{\rho}\left(\rho\right) + g\left(\rho\right)^{-1} g''_{\rho}\left(\rho\right)\Big)\Bigg) \\
= \; & \frac{1}{2} \exp\left(-\nu g_{\nu}\left(0\right)\right) \Big(-\nu g''_{\nu}\left(0\right) g_{\nu}\left(0\right) - \nu\left(g'_{\nu}\left(0\right)\right)^{2} \qquad (257) \\
& - \nu g''_{\nu}\left(0\right) g\left(0\right)^{-1} g_{\rho}\left(0\right) + \nu g'_{\nu}\left(0\right) g\left(0\right)^{-2} g'\left(0\right) g_{\rho}\left(0\right) \\
& - \nu g'_{\nu}\left(0\right) g\left(0\right)^{-1} g'_{\rho}\left(0\right) - \nu g''_{\nu}\left(0\right) g_{\nu}\left(0\right) - \nu g'_{\nu}\left(0\right) g'_{\nu}\left(0\right) \\
& - \nu g''_{\nu}\left(0\right) g\left(0\right)^{-1} g_{\rho}\left(0\right) + \nu g'_{\nu}\left(0\right) g\left(0\right)^{-2} g'\left(0\right) g_{\rho}\left(0\right) \\
& - \nu g'_{\nu}\left(0\right) g\left(0\right)^{-1} g'_{\rho}\left(0\right) + g''_{\nu}\left(0\right) + 2g\left(0\right)^{-3} \left(g'\left(0\right)\right)^{2} g_{\rho}\left(0\right) \\
& - g\left(0\right)^{-2} g''\left(0\right) g_{\rho}\left(0\right) - 2g\left(0\right)^{-2} g'\left(0\right) g'_{\rho}\left(0\right) + g\left(0\right)^{-1} g''_{\rho}\left(0\right)\Big) \\
= \; & \frac{1}{2} \exp\left(-\nu g_{\nu}\left(0\right)\right) \Big(-\nu g''_{\nu}\left(0\right) g_{\nu}\left(0\right) - \nu\left(g'_{\nu}\left(0\right)\right)^{2} - \nu g''_{\nu}\left(0\right) g_{\rho}\left(0\right) \qquad (258) \\
& + \nu g'_{\nu}\left(0\right) g'\left(0\right) g_{\rho}\left(0\right) - \nu g'_{\nu}\left(0\right) g'_{\rho}\left(0\right) \\
& - \nu g''_{\nu}\left(0\right) g_{\nu}\left(0\right) - \nu g'_{\nu}\left(0\right) g'_{\nu}\left(0\right) - \nu g''_{\nu}\left(0\right) g_{\rho}\left(0\right) + \nu g'_{\nu}\left(0\right) g'\left(0\right) g_{\rho}\left(0\right) \\
& - \nu g'_{\nu}\left(0\right) g'_{\rho}\left(0\right) + g''_{\nu}\left(0\right) + 2\left(g'\left(0\right)\right)^{2} g_{\rho}\left(0\right) - g''\left(0\right) g_{\rho}\left(0\right) \\
& - 2g'\left(0\right) g'_{\rho}\left(0\right) + g''_{\rho}\left(0\right)\Big) \\
= \; & \frac{1}{2} \exp\left(-\nu g_{\nu}\left(0\right)\right) \Big(-2\nu g''_{\nu}\left(0\right) g_{\nu}\left(0\right) - 2\nu\left(g'_{\nu}\left(0\right)\right)^{2} - 2\nu g''_{\nu}\left(0\right) g_{\rho}\left(0\right) \qquad (259) \\
& + 2\nu g'_{\nu}\left(0\right) g'\left(0\right) g_{\rho}\left(0\right) - 2\nu g'_{\nu}\left(0\right) g'_{\rho}\left(0\right) \\
& + g''_{\nu}\left(0\right) + 2\left(g'\left(0\right)\right)^{2} g_{\rho}\left(0\right) - g''\left(0\right) g_{\rho}\left(0\right) \\
& - 2g'\left(0\right) g'_{\rho}\left(0\right) + g''_{\rho}\left(0\right)\Big) \\
= \; & \frac{1}{2} \exp\left(-\nu g_{\nu}\left(0\right)\right) \left(g''_{\nu}\left(0\right)\left(-2\nu g_{\nu}\left(0\right) - 2\nu g_{\rho}\left(0\right) + 1\right)\right. \qquad (260) \\
& + \nu g'_{\nu}\left(0\right) \Big(-2g'_{\nu}\left(0\right) + 2g'\left(0\right) g_{\rho}\left(0\right) - 2g'_{\rho}\left(0\right)\Big) \\
& + 2\left(g'\left(0\right)\right)^{2} g_{\rho}\left(0\right) - g''\left(0\right) g_{\rho}\left(0\right) \\
& - 2g'\left(0\right) g'_{\rho}\left(0\right) + g''_{\rho}\left(0\right)\Big)
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \frac{1}{2}\exp\left(-\nu\left(-\delta\,\ln x_1-(1-\delta)\,\ln x_2\right)\right) \\
& \left(\left(-\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1-\ln x_2\right)^3\right)\right. \\
& \left(-2\nu\left(-\delta\,\ln x_1-(1-\delta)\,\ln x_2\right)-2\nu\left(\delta\,\ln x_1+(1-\delta)\,\ln x_2\right)+1\right) \\
& +\nu\frac{\delta\left(1-\delta\right)}{2}\left(\ln x_1-\ln x_2\right)^2\left(-2\frac{\delta\left(1-\delta\right)}{2}\left(\ln x_1-\ln x_2\right)^2\right. \\
& +2\left(-\delta\,\ln x_1-(1-\delta)\,\ln x_2\right)\left(\delta\,\ln x_1+(1-\delta)\,\ln x_2\right) \\
& \left.\left.-2\left(-\delta\left(\ln x_1\right)^2-(1-\delta)\left(\ln x_2\right)^2\right)\right) \\
& +2\left(-\delta\,\ln x_1-(1-\delta)\,\ln x_2\right)^2\left(\delta\,\ln x_1+(1-\delta)\,\ln x_2\right) \\
& -\left(\delta\,\left(\ln x_1\right)^2+(1-\delta)\,\left(\ln x_2\right)^2\right)\left(\delta\,\ln x_1+(1-\delta)\,\ln x_2\right) \\
& -2\left(-\delta\,\ln x_1-(1-\delta)\,\ln x_2\right)\left(-\delta\left(\ln x_1\right)^2-(1-\delta)\left(\ln x_2\right)^2\right) \\
& \left.+\delta\left(\ln x_1\right)^3+(1-\delta)\left(\ln x_2\right)^3\right)
\end{aligned}
\tag{261}
$$

$$
\begin{aligned}
= \quad & \frac{1}{2}x_1^{\nu\delta}x_2^{\nu(1-\delta)}\left(-\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1-\ln x_2\right)^3\right. \\
& \left(2\nu\delta\,\ln x_1+2\nu\left(1-\delta\right)\,\ln x_2-2\nu\delta\,\ln x_1-2\nu\left(1-\delta\right)\,\ln x_2+1\right) \\
& +\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\left(\ln x_1\right)^2-2\ln x_1\ln x_2+\left(\ln x_2\right)^2\right) \\
& \left(-\delta\left(1-\delta\right)\left(\ln x_1\right)^2+2\delta\left(1-\delta\right)\ln x_1\ln x_2-\delta\left(1-\delta\right)\left(\ln x_2\right)^2\right. \\
& -2\delta^2\left(\ln x_1\right)^2-4\delta\left(1-\delta\right)\ln x_1\,\ln x_2-2\left(1-\delta\right)^2\left(\ln x_2\right)^2 \\
& \left.+2\delta\left(\ln x_1\right)^2+2\left(1-\delta\right)\left(\ln x_2\right)^2\right) \\
& +2\delta^3\left(\ln x_1\right)^3+6\delta^2\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2+6\delta\left(1-\delta\right)^2\ln x_1\left(\ln x_2\right)^2 \\
& +2\left(1-\delta\right)^3\left(\ln x_2\right)^3-\delta^2\,\left(\ln x_1\right)^3-\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2 \\
& -\delta\left(1-\delta\right)\ln x_1\left(\ln x_2\right)^2-\left(1-\delta\right)^2\left(\ln x_2\right)^3-2\delta^2\left(\ln x_1\right)^3 \\
& -2\delta\left(1-\delta\right)\ln x_1\left(\ln x_2\right)^2-2\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2-2\left(1-\delta\right)^2\left(\ln x_2\right)^3 \\
& \left.+\delta\left(\ln x_1\right)^3+\left(1-\delta\right)\left(\ln x_2\right)^3\right)
\end{aligned}
\tag{262}
$$

$$
\begin{aligned}
= \quad & \frac{1}{2}x_1^{\nu\delta}x_2^{\nu(1-\delta)}\left(-\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1-\ln x_2\right)^3\right. \\
& +\left(\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_1\right)^2-\nu\delta\left(1-\delta\right)\ln x_1\ln x_2+\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_2\right)^2\right) \\
& \left(\left(-\delta\left(1-\delta\right)-2\delta^2+2\delta\right)\left(\ln x_1\right)^2\right. \\
& +\left(2\delta\left(1-\delta\right)-4\delta\left(1-\delta\right)\right)\ln x_1\ln x_2 \\
& \left.+\left(-\delta\left(1-\delta\right)-2\left(1-\delta\right)^2+2\left(1-\delta\right)\right)\left(\ln x_2\right)^2\right) \\
& \left.+\left(2\delta^3-\delta^2-2\delta^2+\delta\right)\left(\ln x_1\right)^3\right.
\end{aligned}
\tag{263}
$$

$$
\begin{aligned}
&+\left(6\delta^2\left(1-\delta\right)-\delta\left(1-\delta\right)-2\delta\left(1-\delta\right)\right)\left(\ln x_1\right)^2\ln x_2\\
&+\left(6\delta\left(1-\delta\right)^2-\delta\left(1-\delta\right)-2\delta\left(1-\delta\right)\right)\ln x_1\left(\ln x_2\right)^2\\
&+\left(2\left(1-\delta\right)^3-\left(1-\delta\right)^2-2\left(1-\delta\right)^2+\left(1-\delta\right)\right)\left(\ln x_2\right)^3\Big)
\end{aligned}
$$

$$
\begin{aligned}
= \;& \frac{1}{2}\,x_1^{\nu\delta}x_2^{\nu(1-\delta)}\left(-\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1-\ln x_2\right)^3\right. \qquad (264)\\
&+\left(\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_1\right)^2-\nu\delta\left(1-\delta\right)\ln x_1\ln x_2+\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_2\right)^2\right)\\
&\Big(\left(-\delta+\delta^2-2\delta^2+2\delta\right)\left(\ln x_1\right)^2\\
&-2\delta\left(1-\delta\right)\ln x_1\ln x_2\\
&+\left(-\delta+\delta^2-2+4\delta-2\delta^2+2-2\delta\right)\left(\ln x_2\right)^2\Big)\\
&+\left(2\delta^3-3\delta^2+\delta\right)\left(\ln x_1\right)^3\\
&+\left(6\delta^2-6\delta^3-\delta+\delta^2-2\delta+2\delta^2\right)\left(\ln x_1\right)^2\ln x_2\\
&+\left(6\delta-12\delta^2+6\delta^3-\delta+\delta^2-2\delta+2\delta^2\right)\ln x_1\left(\ln x_2\right)^2\\
&+\left.\left(2-6\delta+6\delta^2-2\delta^3-1+2\delta-\delta^2-2+4\delta-2\delta^2+1-\delta\right)\left(\ln x_2\right)^3\right)
\end{aligned}
$$

$$
\begin{aligned}
= \;& \frac{1}{2}\,x_1^{\nu\delta}x_2^{\nu(1-\delta)}\left(-\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1-\ln x_2\right)^3\right. \qquad (265)\\
&+\left(\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_1\right)^2-\nu\delta\left(1-\delta\right)\ln x_1\ln x_2+\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_2\right)^2\right)\\
&\Big(\left(-\delta^2+\delta\right)\left(\ln x_1\right)^2-2\delta\left(1-\delta\right)\ln x_1\ln x_2+\left(-\delta^2+\delta\right)\left(\ln x_2\right)^2\Big)\\
&+\left(2\delta^3-3\delta^2+\delta\right)\left(\ln x_1\right)^3+\left(-6\delta^3+9\delta^2-3\delta\right)\left(\ln x_1\right)^2\ln x_2\\
&+\left.\left(6\delta^3-9\delta^2+3\delta\right)\ln x_1\left(\ln x_2\right)^2+\left(-2\delta^3+3\delta^2-\delta\right)\left(\ln x_2\right)^3\right)
\end{aligned}
$$

$$
\begin{aligned}
= \;& \frac{1}{2}\,x_1^{\nu\delta}x_2^{\nu(1-\delta)}\left(-\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^3\right. \qquad (266)\\
&+\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_1\right)^2\ln x_2-\left(1-2\delta\right)\delta\left(1-\delta\right)\ln x_1\ \left(\ln x_2\right)^2\\
&+\frac{1}{3}\left(1-2\delta\right)\delta\left(1-\delta\right)\left(\ln x_2\right)^3\\
&+\left(\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_1\right)^2-\nu\delta\left(1-\delta\right)\ln x_1\ln x_2+\frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_2\right)^2\right)\\
&\Big(\delta\left(1-\delta\right)\left(\ln x_1\right)^2-2\delta\left(1-\delta\right)\ln x_1\ln x_2+\delta\left(1-\delta\right)\left(\ln x_2\right)^2\Big)\\
&+\delta\left(1-\delta\right)\left(1-2\delta\right)\left(\ln x_1\right)^3-3\delta\left(1-\delta\right)\left(1-2\delta\right)\left(\ln x_1\right)^2\ln x_2\\
&+\left.3\delta\left(1-\delta\right)\left(1-2\delta\right)\ln x_1\left(\ln x_2\right)^2-\delta\left(1-\delta\right)\left(1-2\delta\right)\left(\ln x_2\right)^3\right)
\end{aligned}
$$

$$
\begin{aligned}
= \; & \frac{1}{2} x_1^{\nu\delta} x_2^{\nu(1-\delta)} \left(\frac{1}{2} \nu\delta^2 (1-\delta)^2 (\ln x_1)^4 - \nu\delta^2 (1-\delta)^2 (\ln x_1)^3 \ln x_2 \right. \\
& + \frac{1}{2} \nu\delta^2 (1-\delta)^2 (\ln x_1)^2 (\ln x_2)^2 - \nu\delta^2 (1-\delta)^2 (\ln x_1)^3 \ln x_2 \\
& + 2\nu\delta^2 (1-\delta)^2 (\ln x_1)^2 (\ln x_2)^2 - \nu\delta^2 (1-\delta)^2 \ln x_1 (\ln x_2)^3 \\
& + \frac{1}{2} \nu\delta^2 (1-\delta)^2 (\ln x_1)^2 (\ln x_2)^2 - \nu\delta^2 (1-\delta)^2 \ln x_1 (\ln x_2)^3 \\
& + \frac{1}{2} \nu\delta^2 (1-\delta)^2 (\ln x_2)^4 \\
& + \left(-\frac{1}{3} (1-2\delta) \delta (1-\delta) + \delta (1-\delta) (1-2\delta) \right) (\ln x_1)^3 \\
& + \left((1-2\delta) \delta (1-\delta) - 3\delta (1-\delta) (1-2\delta) \right) (\ln x_1)^2 \ln x_2 \\
& + \left(- (1-2\delta) \delta (1-\delta) + 3\delta (1-\delta) (1-2\delta) \right) \ln x_1 (\ln x_2)^2 \\
& \left. + \left(\frac{1}{3} (1-2\delta) \delta (1-\delta) - \delta (1-\delta) (1-2\delta) \right) (\ln x_2)^3 \right)
\end{aligned} \tag{267}
$$

$$
\begin{aligned}
= \; & \frac{1}{2} x_1^{\nu\delta} x_2^{\nu(1-\delta)} \left(\frac{1}{2} \nu\delta^2 (1-\delta)^2 (\ln x_1)^4 - 2\nu\delta^2 (1-\delta)^2 (\ln x_1)^3 \ln x_2 \right. \\
& + 3\nu\delta^2 (1-\delta)^2 (\ln x_1)^2 (\ln x_2)^2 \\
& - 2\nu\delta^2 (1-\delta)^2 \ln x_1 (\ln x_2)^3 + \frac{1}{2} \nu\delta^2 (1-\delta)^2 (\ln x_2)^4 \\
& + \frac{2}{3} \delta (1-\delta) (1-2\delta) (\ln x_1)^3 - 2\delta (1-\delta) (1-2\delta) (\ln x_1)^2 \ln x_2 \\
& \left. + 2\delta (1-\delta) (1-2\delta) \ln x_1 (\ln x_2)^2 - \frac{2}{3} \delta (1-\delta) (1-2\delta) (\ln x_2)^3 \right)
\end{aligned} \tag{268}
$$

$$
\begin{aligned}
= \; & \frac{1}{2} x_1^{\nu\delta} x_2^{\nu(1-\delta)} \left(\frac{1}{2} \nu\delta^2 (1-\delta)^2 \left((\ln x_1)^4 - 4 (\ln x_1)^3 \ln x_2 \right) \right. \\
& + 6 (\ln x_1)^2 (\ln x_2)^2 - 4 \ln x_1 (\ln x_2)^3 + (\ln x_2)^4 \\
& + \frac{2}{3} \delta (1-\delta) (1-2\delta) \\
& \left. \left((\ln x_1)^3 - 3 (\ln x_1)^2 \ln x_2 + 3 \ln x_1 (\ln x_2)^2 - (\ln x_2)^3 \right) \right)
\end{aligned} \tag{269}
$$

$$
\begin{aligned}
= \; & \delta (1-\delta) \; x_1^{\nu\delta} x_2^{\nu(1-\delta)} \\
& \left(\frac{1}{3} (1-2\delta) (\ln x_1 - \ln x_2)^3 + \frac{1}{4} \nu\delta (1-\delta) (\ln x_1 - \ln x_2)^4 \right)
\end{aligned} \tag{270}
$$

Hence, we can approximate $\partial y / \partial \rho$ by a second-order Taylor series approximation:

$$
\frac{\partial y}{\partial \rho} \approx \gamma\nu \left(f_\rho (0) + \rho \, f'_\rho (0) \right) \tag{271}
$$

$$
= -\frac{1}{2} \gamma\nu\delta (1-\delta) \; x_1^{\nu\delta} \, x_2^{\nu(1-\delta)} (\ln x_1 - \ln x_2)^2 \tag{272}
$$

$$
\begin{aligned}
& + \gamma\nu\rho\delta\left(1-\delta\right)\, x_1^{\nu\delta} x_2^{\nu(1-\delta)} \\
& \left(\frac{2}{3}\left(1-2\delta\right)\left(\ln x_1 - \ln x_2\right)^3 + \frac{1}{2}\nu\delta\left(1-\delta\right)\left(\ln x_1 - \ln x_2\right)^4\right) \\
= \;& \gamma\nu\delta\left(1-\delta\right)\, x_1^{\nu\delta}\, x_2^{\nu(1-\delta)}\left(-\frac{1}{2}\left(\ln x_1 - \ln x_2\right)^2\right. \qquad (273) \\
& \left. + \frac{1}{3}\rho\left(1-2\delta\right)\left(\ln x_1 - \ln x_2\right)^3 + \frac{1}{4}\rho\nu\delta\left(1-\delta\right)\left(\ln x_1 - \ln x_2\right)^4\right)
\end{aligned}
$$

B. Script for Monte Carlo Simulation

```
# load the micEconCES package
library( micEconCES )

# seed for random number generation
set.seed( 1234 )

# number of replications
nRep <- 1000

# number of observations
nObs <- 100

# rho
rho <- 1/3

# variance of the error term
uStdDev <- 1.5

# create data set with explanatory variables
cesData <- data.frame( x1 = rchisq( nObs, 10 ), x2 = rchisq( nObs, 10 ) )

# names of explanatory variables
xxNames <- c( "x1", "x2" )

# variable returns to scale
vrs <- TRUE # FALSE #

# coefficients
cesCoef <- c( gamma = 1, delta = 0.6, rho = rho, nu = 1.1 )[ 1:( 3 + vrs ) ]

# calculate deterministic endogenous variable
cesData$yd <- cesCalc( xNames = xxNames, data = cesData, coef = cesCoef )

# estimation methods
allMethods <- c( "Kmenta", "nls", "LM", "CG", "Newton", "BFGS",
   "Nelder-Mead", "SANN", "DE", "L-BFGS-B", "PORT" )

# objects to store the results
estCoef <- array( NA,
   dim = c( nRep, length( cesCoef ), length( allMethods ) ),
   dimnames = list( 1:nRep, names( cesCoef ), allMethods ) )
convergence <- estCoef[ , 1, ]
rss <- estCoef[ , 1, ]
rSquared <- estCoef[ , 1, ]
iterations <- estCoef[ , 1, ]
```

```
## start the monte carlo experiment
for( i in 1:nRep ) {
   cat( i, ":␣", sep = "" )
   ptm <- proc.time()

   # adding noise to the endogenous variable
   repeat{
      cesData$ys <- cesData$yd + rnorm( nObs, sd = uStdDev )
      if( min( cesData$ys ) > 0 ) {
         break
      } else {
         cat( "#" )
      }
   }

   # estimate the model using different estimation methods
   for( method in allMethods ) {
      extraArgs <- list()
      if( method == "nls" ) {
         extraArgs <- list(
            control = nls.control( warnOnly = TRUE ) )
      } else if( method == "LM" ) {
         extraArgs <- list(
            control = nls.lm.control( maxiter = 250 ) )
      } else if( method == "Newton" ) {
         extraArgs <- list( iterlim = 250 )
      } else if( method %in% c( "BFGS", "L-BFGS-B" ) ) {
         extraArgs <- list( control = list( maxit = 250 ) )
      } else if( method == "CG" ) {
         extraArgs <- list(
            control = list( maxit = 1000, reltol = 1e-4, type = 2 ) )
      } else if( method == "SANN" ) {
         extraArgs <- list( control = list( maxit = 50000 ) )
      } else if( method == "DE" ) {
         extraArgs <- list(
            control = DEoptim.control( trace = FALSE, itermax = 1000 ) )
      }
      allArgs <- c( list( yName = "ys", xNames = xxNames, data = cesData,
         method = method, vrs = vrs ), extraArgs )
      cesResult <- try( do.call( "cesEst", allArgs ) )
      if( class( cesResult )[1] != "try-error" ) {
         # store the estimated coefficients
         estCoef[ i, , method ] <- coef( cesResult )
         # store if the estimation has converged
         if( !is.null( cesResult$convergence ) ) {
            convergence[ i, method ] <- cesResult$convergence
         }
```

```
         # sum of squared residuals
         rss[ i, method ] <- cesResult$rss
         # R-squared values
         rSquared[ i, method ] <- summary( cesResult )$r.squared
         # number of iterations
         if( !is.null( cesResult$iter ) ) {
            iterations[ i, method ] <- sum( cesResult$iter )
         }
      }
   }

   ptmNew <- proc.time()
   cat( ptmNew - ptm, "\n" )
   ptm <- ptmNew
}

############ calculate summary results ###############
# differences between the estimated and the true coefficients
diffCoef <- estCoef - aperm(
   array( cesCoef, dim = c( length( cesCoef ), length( allMethods ), nRep ) ),
   c( 3, 1, 2 ) )

# elasticities of substitution and difference between estimates and true value
estSigma <- 1 / ( 1 + estCoef[ ,"rho", ] )
diffSigma <- estSigma - 1 / ( 1 + rho )

# function to calculate summary results of the Monte Carlo simulation
# depending on the selection of replications
calcMcResults <- function( repSelect ) {

   result <- list()

   # biases of the estimated coefficients and elasticity of substitution
   result$bias <- colMeans( diffCoef[ repSelect, , ] )
   # all.equal( bias, colMeans( estCoef ) - matrix( cesCoef, nrow = length( cesCoef ), ncol = length( allMeth
   result$bias <- rbind( result$bias,
      sigma = colMeans( diffSigma[ repSelect, ] ) )

   # median deviation of estimated coef. and elast. of subst. from their true values
   result$devMed <- colMedians( diffCoef[ repSelect, , ] )
   result$devMed <- rbind( result$devMed,
      sigma = colMedians( diffSigma[ repSelect, ] ) )

   # root mean squared errors of the estimated coefficients and elasticity of substitution
   result$rmse <- sqrt( colSums( diffCoef[ repSelect, , ]^2 ) / nRep )
   result$rmse <- rbind( result$rmse,
      sigma =  colSums( diffSigma[ repSelect, ]^2 ) / nRep )
```

```
   # mean absolute deviations
   result$mad <- colMeans( abs( diffCoef[ repSelect, , ] ) )
   result$mad <- rbind( result$mad,
      sigma = colMeans( abs( diffSigma[ repSelect, ] ) ) )

   # median absolute deviations
   result$adMed <- colMedians( abs( diffCoef[ repSelect, , ] ) )
   result$adMed <- rbind( result$adMed,
      sigma = colMedians( abs( diffSigma[ repSelect, ] ) ) )

   # mean RSS
   result$rssMean <- colMeans( rss[ repSelect, ] )

   # mean R-squared values
   result$rSquaredMean <- colMeans( rSquared[ repSelect, ] )

   return( result )
}

# summary results of *all* replications
resultAll <- calcMcResults( 1:nRep )

# summary results of replications without errors (in any method)
resultNoErr <- calcMcResults( rowSums( is.na( rss ) ) == 0 )

# summary results of replications without errors or non-convergence (in any method)
resultConv <- calcMcResults(
   rowSums( is.na( rss ) | ( !convergence & !is.na( convergence ) ) ) == 0 )


############ create tables for the paper ###############
# general results
tabGeneral <- data.frame( nNoConv =
   colSums( is.na( rss ) | ( !convergence & !is.na( convergence ) ) ) )
tabGeneral$nConv <-
   colSums( !is.na( rss ) & ( convergence | is.na( convergence ) ) )
tabGeneral$rssAll <- resultAll$rssMean
tabGeneral$rssConv <- resultConv$rssMean


############# write tables to disk ###############
library( xtable )
# general results
xTabGeneral <- xtable( tabGeneral, digits = c( rep( 0, 3 ), rep( 7, 2 ) ),
   align = c( "l", rep( "r", 4 ) ) )
print( xTabGeneral, file = "../tables/mcGeneral.tex", floating = FALSE )
```

```
# bias
xBias <- xtable( t( resultAll$bias ), digits = rep( 5, 6 ),
   align = c( "l", rep( "r", 5 ) ) )
print( xBias, file = "../tables/mcBias.tex", floating = FALSE )

# root mean square error
xRmse <- xtable( t( resultAll$rmse ), digits = rep( 5, 6 ),
   align = c( "l", rep( "r", 5 ) ) )
print( xRmse, file = "../tables/mcRmse.tex", floating = FALSE )
```

References

Ali MM, Törn A (2004). "Population set-based global optimization algorithms: some modifications and numerical studies." *Computers & Operations Research*, **31**(10), 1703–1725.

Amras P (2004). "Is the U.S. Aggregate Production Function Cobb-Douglas? New Estimates of the Elasticity of Substitution." *Contribution in Macroeconomics*, **4**(1), Article 4.

Ardia D, Mullen K (2009). *DEoptim: Global Optimization by Differential Evolution*. R package version 2.0-3, URL http://CRAN.R-project.org/package=DEoptim.

Arrow KJ, Chenery BH, Minhas BS, Solow RM (1961). "Capital-labor substitution and economic efficiency." *The Review of Economics and Statistics*, **43**(3), 225–250. URL http://www.jstor.org/stable/1927286.

Beale EML (1972). "A Derivation of Conjugate Gradients." In FA Lootsma (ed.), *Numerical Methods for Nonlinear Optimization*, pp. 39–43. Academic Press, London.

Bélisle CJP (1992). "Convergence Theorems for a Class of Simulated Annealing Algorithms on R^d." *Journal of Applied Probability*, **29**, 885–895.

Bentolila SJ, Gilles SP (2006). "Explaining Movements in the Labour Share." *Contributions to Macroeconomics*, **3**(1), Article 9.

Broyden CG (1970). "The Convergence of a Class of Double-rank Minimization Algorithms." *Journal of the Institute of Mathematics and Its Applications*, **6**, 76–90.

Byrd R, Lu P, Nocedal J, Zhu C (1995). "A limited memory algorithm for bound constrained optimization." *SIAM Journal for Scientific Computing*, **16**, 1190–1208.

Caselli F (2005). "Accounting for Cross-Country Income Differences." In P Aghion, SN Durlauf (eds.), *Handbook of Economic Growth*, pp. 679–742. North Holland.

Caselli F, Coleman Wilbur John I (2006). "The World Technology Frontier." *American Economic Review*, **96**(3), 499–522. URL http://www.jstor.org/stable/30034059.

Cerny V (1985). "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm." *Journal of Optimization Theory and Applications*, **45**, 41–51.

Chambers JM, Hastie TJ (1992). *Statistical Models in S*. Chapman & Hall, London.

Dennis JE, Schnabel RB (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs (NJ, USA).

Elzhov TV, Mullen KM (2009). *minpack.lm: R Interface to the Levenberg-Marquardt Nonlinear Least-Squares Algorithm Found in MINPACK*. R package version 1.1-4, URL http://CRAN.R-project.org/package=minpack.lm.

Fletcher R (1970). "A New Approach to Variable Metric Algorithms." *Computer Journal*, **13**, 317–322.

Fletcher R, Reeves C (1964). "Function minimization by conjugate gradients." *Computer Journal*, **7**, 48–154.

Fox J (2009). *car: Companion to Applied Regression*. R package version 1.2-16, URL http://CRAN.R-project.org/package=car.

Gay DM (1990). "Usage Summary for Selected Optimization Routines." *Computing Science Technical Report 153*, AT&T Bell Laboratories. URL http://netlib.bell-labs.com/cm/cs/cstr/153.pdf.

Goldfarb D (1970). "A Family of Variable Metric Updates Derived by Variational Means." *Mathematics of Computation*, **24**, 23–26.

Greene WH (2008). *Econometric Analysis*. 6 edition. Prentice Hall.

Henningsen A (2009). *micEcon: Tools for Microeconomic Analysis and Microeconomic Modeling*. R package version 0.6, URL http://CRAN.R-project.org/package=micEcon.

Henningsen A, Hamann JD (2007). "systemfit: A Package for Estimating Systems of Simultaneous Equations in R." *Journal of Statistical Software*, **23**(4), 1–40. URL http://www.jstatsoft.org/v23/i04/.

Henningsen A, Henningsen G (2010). *micEconCES: Analysis with the Constant Elasticity of Scale (CES) Function*. R package version 0.6, URL http://CRAN.R-project.org/package=micEconCES.

Hoff A (2004). "The Linear Approximation of the CES Function with n Input Variables." *Marine Resource Economics*, **19**, 295–306.

Kelley CT (1999). *Iterative Methods of Optimization*. SIAM Society for Industrial and Applied Mathematics, Philadelphia.

Kirkpatrick S, Gelatt CD, Vecchi MP (1983). "Optimization by Simulated Annealing." *Science*, **220**(4598), 671–680.

Kmenta J (1967). "On Estimation of the CES Production Function." *International Economic Review*, **8**, 180–189.

Maddala G, Kadane J (1967). "Estimation of Returns to Scale and the Elasticity of Substitution." *Econometrica*, **24**, 419–423.

Marquardt DW (1963). "An Algorithm for Least-Squares Estimation of Non-linear Parameters." *Journal of the Society for Industrial and Applied Mathematics*, **11**(2), 431–441. URL http://www.jstor.org/stable/2098941.

Mishra SK (2007). "A Note on Numerical Estimation of Sato's Two-Level CES Production Function." *MPRA Paper 1019*, North-Eastern Hill University, Shillong.

Moré JJ, Garbow BS, Hillstrom KE (1980). *MINPACK*. Argonne National Laboratory. URL http://www.netlib.org/minpack/.

Nelder JA, Mead R (1965). "A Simplex Algorithm for Function Minimization." *Computer Journal*, **7**, 308–313.

Polak E, Ribière G (1969). "Note sur la convergence de méthodes de directions conjuguées." *Revue Francaise d'Informatique et de Recherche Opérationnelle*, **16**, 35–43.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Schnabel RB, Koontz JE, Weiss BE (1985). "A Modular System of Algorithms for Unconstrained Minimization." *ACM Transactions on Mathematical Software*, **11**(4), 419–440.

Shanno DF (1970). "Conditioning of Quasi-Newton Methods for Function Minimization." *Mathematics of Computation*, **24**, 647–656.

Soda G, Vichi EG (1976). "Least Squares Estimation of C.E.S. Production Function's Non-linear Parameters." In *Proceedings of the eighth international conference on APL*, pp. 408 – 413. ACM New York, NY, USA.

Sorenson HW (1969). "Comparison of Some Conjugate Direction Procedures for Function Minimization." *Journal of the Franklin Institute*, **288**(6), 421–441.

Storn R, Price K (1997). "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." *Journal of Global Optimization*, **11**, 341–359.

Thursby JG (1980). "CES Estimation Techniques." *The Review of Economics and Statistics*, **62**, 295–299.

Thursby JG, Lovell CAK (1978). "In Investigation of the Kmenta Approximation to the CES Function." *International Economic Review*, **19**(2), 363–377.

Uebe G (2000). "Kmenta Approximation of the CES Production Function." Macromoli: Goetz Uebe's notebook on macroeconometric models and literature, http://www2.hsu-hh.de/uebe/Lexikon/K/Kmenta.pdf [accessed 2010-02-09].

Affiliation:

Géraldine Henningsen
Danmarks Statistik
E-mail: geraldine.henningsen@gmail.com

Arne Henningsen
Institute of Food and Resource Economics
University of Copenhagen
Rolighedsvej 25
1958 Frederiksberg C, Denmark
E-mail: arne.henningsen@gmail.com
URL: http://www.arne-henningsen.name/