Computable General Equilibrium Models: Production function

Wolfgang Britz, Institute for Agricultural Policy,
University Bonn

Simulation Models for Policy Analysis

Summer Term 2020

Aims for today

- Refresh your memory about the CES, CD and Leontief production function
- Learn how production is depicted in CGEs
- Learn how the CES function is implemented in a CGE and how the calibration works
- Look at code

Production function I

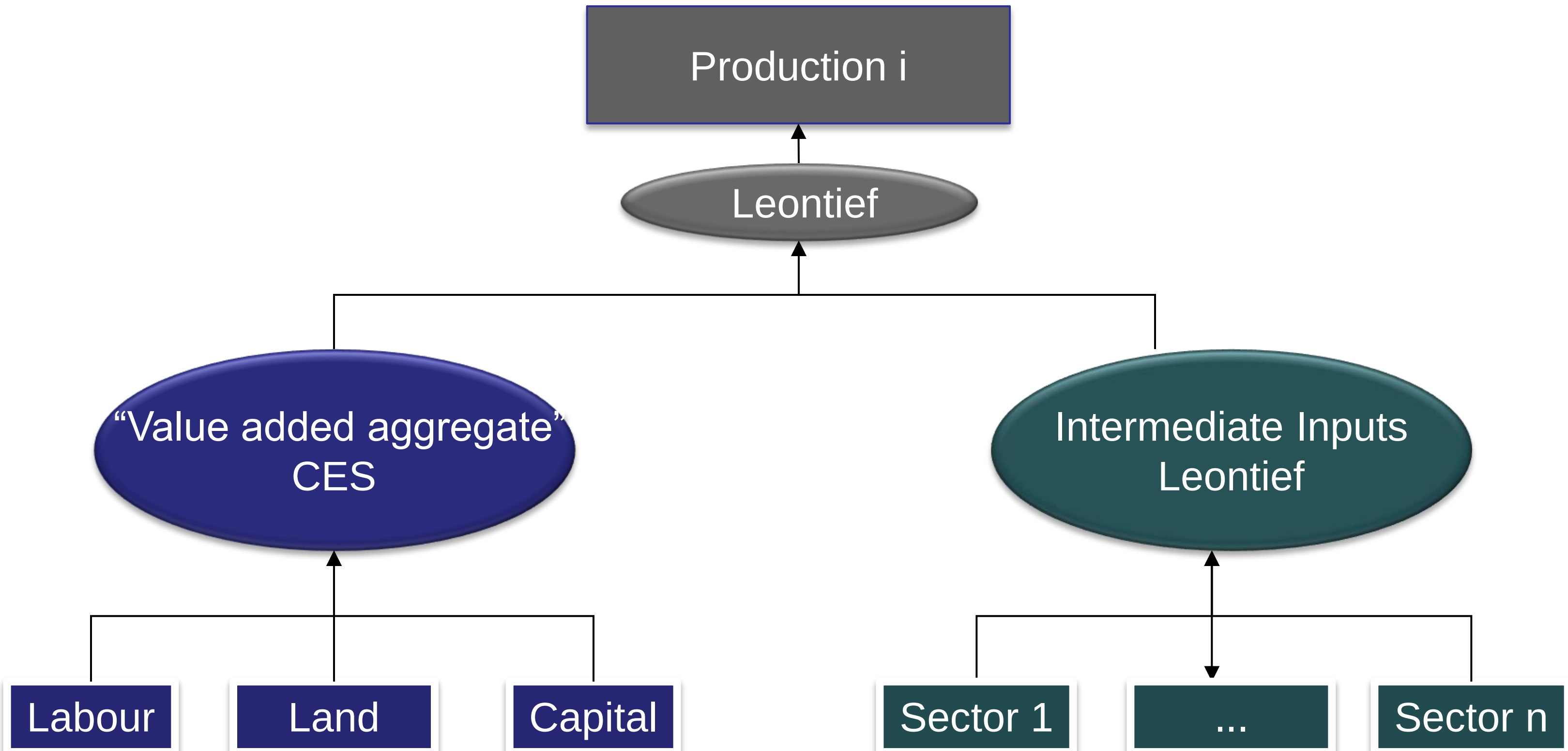
- Typical assumptions in CGEs:
  - Cost minimizing behavior
  - Competitive equilibrium <=> zero marginal profits
  - Constant return to scale
  - (Nested) CES production functions
- Often additionally:
  - Leontief based intermediate input coefficients
  - Leontief bundle between intermediate coefficients and value added nest
  - CES-function for value added nest

Production: CRS

- Constant-return-to-scale:
  - marginal production cost (= input mix) stay constant at given input prices if output quantity changes
  - Consequence: marginal = average production cost
  - marginal revenue (= price in competitive market) = average production cost
  - Zero profit, not only zero marginal profit

  ⇒ Production output at given prices is not defined by profit maximization!
- Total output quantity defined instead
  - by demand for output
  - and/or input supply
  - and/or price feedback in input/output markets

Production function: Example



CES production function

$$y = \beta \left[\sum_i \alpha_i \ x_i^{-\rho} \right]^{\frac{\gamma}{\rho}}$$

- Substitution elasticities are constant

$\sigma = 1/(1+\rho)$ = Fix = (relative change in quantity relation) in relation to (relative change in price relation)

- Remember:

$$\frac{d(x_2/x_1)}{x_2/x_1} = d\ log(x_2/x_1) = d\ log(x_2) - \ d\ log(x_1) = -[\ d\ log(x_1) - \ d\ log(x_2)] = -\frac{d\ (x_1/x_2)}{x_1/x_2}$$

$$\sigma = -\frac{d(x_1/x_2)}{x_1/x_2}\frac{p_1/p_2}{d(p_1/p_2)} = -\frac{d\ \log(x_1/x_2)}{d\ \log(p_1/p_2)}$$

CES production function

$$y = \beta \left[\sum_i \alpha_i \; x_i^{-\rho} \right]^{\frac{\gamma}{\rho}}$$

- Remember:
  - γ describes return-to-scale (1=CRS)
  - β is the shift parameter or Hicks-neutral technical progress multiplier, defines the production frontier
  - Hicks-neutral means that a change in the parameter does not change the composition of inputs at given prices
  - α can be interpreted as cost shares if prices and β are unity in calibration point (so called calibrated share form)

Mnemonics in our model

- Names of variables, equations, parameter follow (closely) ENVISAGE (Environmental Impact and Sustainability Applied General Equilibrium Model)
- Developed at World Bank by Dominique van der Mensbrugghe
- Later introduced at FAO
- Dominique now is now director of GTAP
- Developed with Wolfgang Britz CGEBox which we will use in class
- CGEBox can replicate the GTAP Standard model, but also features from other often-used CGEs

Mnemonics: remember

- We use:
  1. v_... for a variable
  2. p_.. for a parameter
  3. e_.. for a equation
  4. m_.. for a model
  5. .. for a set

Note: Do you remember why do we do that?

CES function in applied modeling

- As we are using a market model, we cannot write min C = …. s.t. x = f(y)

$\Rightarrow$ we need FOC

$\Rightarrow$ These can be written in different ways

$\Rightarrow$ The usual way is presented here, specifically

$\Rightarrow$ Instead of defining the average price as $\tilde{p} = \frac{\sum x_i p_i}{\sum x_i}$, a dual expression is used

CES: Dual price index and demand (FOC)

Composite price index

Input prices

```
v_pva(r,a) =E= sum(f, p_af(r,f,a) * v_pf(r,f)**(1-p_sigmav(r,a)))** ( 1/(1-p_sigmav(r,a)));
```

Cost share

Composite demand
or output quantity (index)

Substitution
elasticity

```
v_xf(r,f,s) =E= p_af(r,f,s) * v_va(r,s) * (v_pva(r,s)/v_pf(r,f))**p_sigmav(r,s);
```

Input demand
quantity (index)

Price relation
Composite price/input price

CES: Calibration

- Use so-called "calibrated share" form:
  1. Set prices to unity:

```
*
* --- prices are unity in benchmark
*       (= SAM values are interpreted as quantity indices)
*
  v_pnd.fx(r,s)    = 1;
  v_pva.fx(r,s)    = 1;
  v_pf.fx(r,f)     = 1;
  v_px.fx(r,c)     = 1;
```

  2. Set quantities to SAM entries (value=quantity index) (e.g.):

```
*
* --- assign factor use per sector
*
  v_xf.l(r,f,s)  = v_sam.l(R,f,s);
```

```
*
* --- calculate value added
*
  v_va.l(r,s)  = sum(f,v_xf.l(r,f,s));
```

  3. Derive share parameters from quantities (e.g.):

```
p_af(r,f,s)     = v_xf.l(r,f,s)/v_va.l(r,s);
```

CES: Calibration

- "Calibrated share" form:
  - Approach works as $1^x = 1$

  => The last term in demand equation becomes unity

```
v_xf(r,f,s) =E= p_af(r,f,s) * v_va(r,s) * (v_pva(r,s)/v_pf(r,f))**p_sigmav(r,s);
```

Calibration point where prices are equal to 1

```
v_xf(r,f,s) =E= p_af(r,f,s) * v_va(r,s)
```

```
p_af(r,f,s)    = v_xf.l(r,f,s)/v_va.l(r,s);
```

CES: Dual price index and FOC

- The same structure with dual price index and FOC demand equation is used to derive:
  - Value added demand (= total factor demand)
  - Composite intermediate demand (= total intermediate demand)
  - Demand for individual factors
  - Demand for individual intermediates

Production block: Price definitions

Zero profits

```
* --- Unit cost definition (net of output tax)
*     Output price (= marginal revenue) = marginal cost = dual price aggregator of top level CES

  e_px(r,c) ..
*
     sum(s_to_c(a,c),p_axp(r,a))*v_px(r,c) =E=

     sum(s_to_c(a,c),(   p_and(r,a)*v_pnd(r,a)**(1-p_sigmap(r,a))
                       + p_ava(r,a)*v_pva(r,a)**(1-p_sigmap(r,a))) ** (1/(1-p_sigmap(r,a))));

* --- Value added price: dual price aggreagator

  e_pva(r,a) ..
*
     v_pva(r,a) =E= sum(f, p_af(r,f,a) * v_pfa(r,f,a)**(1-p_sigmav(r,a)))** ( 1/(1-p_sigmav(r,a)));

* --- Intermediate composite price: dual price aggreagator

  e_pnd(R,s) ..
*
     v_pnd(r,s) =E= sum(c, p_io(r,c,s) *  [v_px(r,c)*(1+p_oTax(r,c))]**(1-p_sigman(r,s)))** ( 1/(1-p_sigman(r,s)));
```

Production block: Input demand

```
* --- Demand for intermediate composite

  e_nd(R,s) ..
*
     v_nd(r,s) =E=  p_and(r,s)* v_x(r,s) * (sum(s_to_c(s,c),v_px(r,c)) / v_pnd(r,s)) ** p_sigmap(r,s)
                     * p_axp(r,s) ** (p_sigmap(r,s)-1);

* --- Demand for value added aggregate

  e_va(R,s) ..
*
     v_va(r,s) =E=  p_ava(r,s)* v_x(r,s) * (sum(s_to_c(s,c),v_px(r,c))/ v_pva(r,s)) ** p_sigmap(r,s)
                     * p_axp(r,s) ** (p_sigmap(r,s)-1);

* --- Factor demand

  e_xf(r,f,a) ..

     v_xf(r,f,a) =E= p_af(r,f,a) * v_va(r,a) * (v_pva(r,a)/v_pfa(r,f,a))**p_sigmav(r,a);

* --- Intermediate demand

  e_xaint(r,c,a) ..

     v_xa(r,c,a) =E= p_io(r,c,a) * v_nd(r,a) * (v_px(r,c)*(1+p_oTax(r,c))/v_pnd(r,a))**p_sigman(r,a);
```

Note:

1. p_axp is ==1 in benchmark, can be used to introduce Hicks-Neutral technical progress
2. Output v_x due to CRS defined by other equations

CES: Leontief as special case

```
v_xf(r,f,s) =E= p_af(r,f,s) * v_va(r,s) * (v_pva(r,s)/v_pf(r,f))**p_sigmav(r,s);
```

Substitution elasticity == 0: $x^0 = 1$

```
v_xf(r,f,s) =E= p_af(r,f,s) * v_va(r,s)
```

Physical input demand is fixed share of output (Leontief)
Input prices do not influence input composition

```
v_pva(r,a) =E= sum(f, p_af(r,f,a) * v_pf(r,f)**(1-p_sigmav(r,a)))** ( 1/(1-p_sigmav(r,a)));
```

Price index is a linear aggregator

=1

=1

```
v_pva(r,a) =E= sum(f, p_af(r,f,a) * v_pf(r,f))
```

CES: CD as special case

Leontief: Substitution elasticity == 1

```
v_xf(r,f,s) =E= p_af(r,f,s) * v_va(r,s) * (v_pva(r,s)/v_pf(r,f))**p_sigmav(r,s);
```

Substitution elasticity == 1: $x^1 = x$

```
v_xf(r,f,s)*v_pf(r,f) =E= p_af(r,f,s) * v_va(r,s) *  v_pva(r,s)
```

Value share (= LHS) is fixed!

```
v_pva(r,a) =E= sum(f, p_af(r,f,a) * v_pf(r,f)**(1-p_sigmav(r,a)))** ( 1/(1-p_sigmav(r,a)));
```

=0

=undefined!

We need another price index!

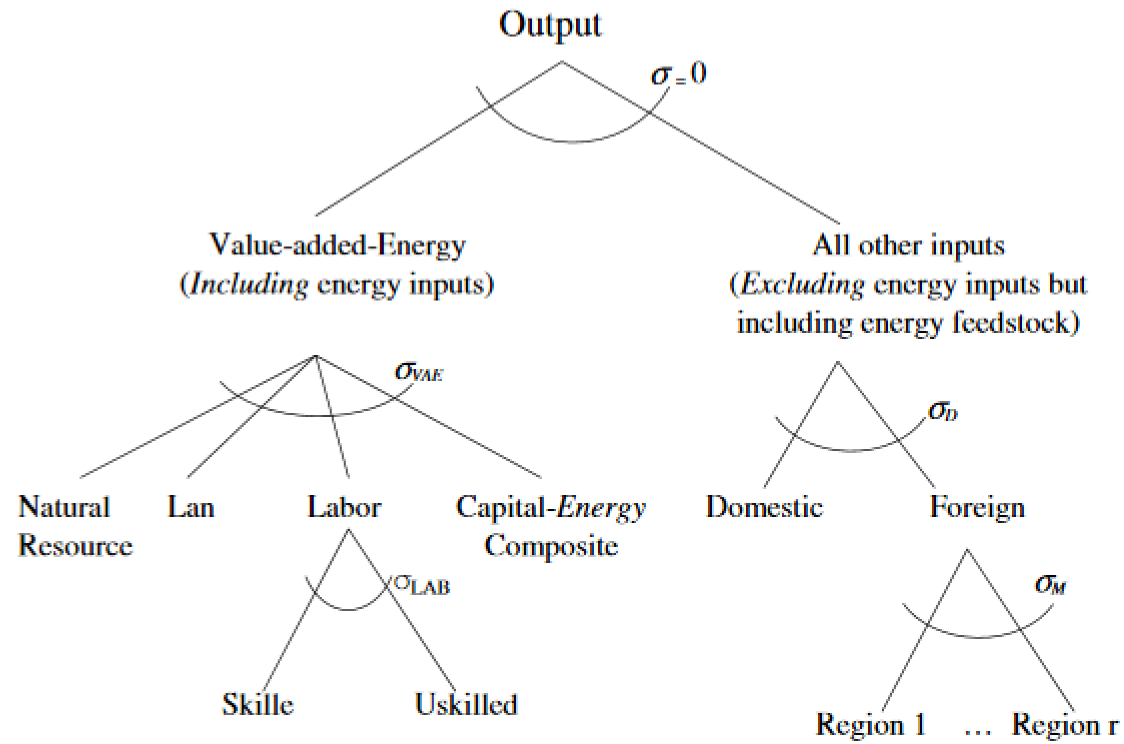
```
v_pva(r,a) =E= sum(f, p_af(r,f,a) * (v_pf(r,f)/p_af(r,f,a))**p_af(r,f,a) )
```

More complex nestings

- Some CGEs use more complex nesting structures, we will use GTAP-E as an example (https://www.gtap.agecon.purdue.edu/resources/res_display.asp?RecordID=923)

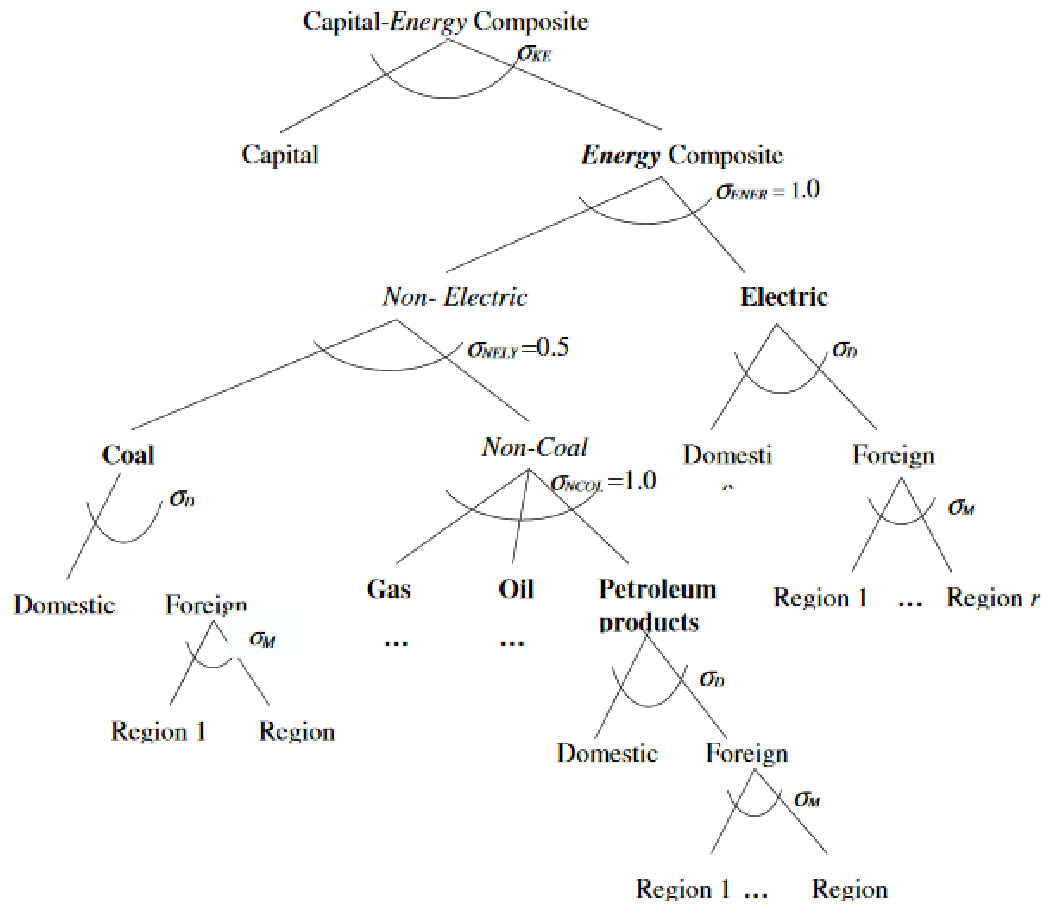
More complex nestings: GTAP-E

Figure 16 GTAP–E Production Structure



More complex nestings: GTAP-E

Figure 17 GTAP–E Capital-Energy Composite Structure



More complex nestings

- Two approaches to implement that in code:
  1. Manually code additional CES nests
  2. Use a generic approach => CGEBox

GTAP-E in CGEBox

```
tNest("CAP+ENE") = YES;
tNest_n_a("VA","CAP+ENE",a)        = YES;
tNest_f_a("CAP+ENE","capital",a) = YES;
tNest_n_a("CAP+ENE","energy",a)   = YES;
sigmaNest(r,"CAP+ENE",a)          = 0.25;

tNest_i_a("energy",ely,a)             = YES;
sigmaNest(r,"energy",a)              = 1.00;

tNest_n_a("energy","non-electric",a)    $ tNest("non-electric")        = YES;
tNest_i_a("energy",coal,a)              $ (not tNest("non-electric")) = YES;
tNest_i_a("energy",nonCoal,a)           $ (not tNest("non-electric")) = YES;

tNest("non-electric") = YES;
tNest_i_a("non-electric",coal,a )      = YES;
tNest_n_a("non-electric","non-coal",a) = YES;
sigmaNest(r,"non-electric",a)          = 0.50;
```

…. Requires generic equations in model for nested CES