ABOUT QUANTITIES AND VALUES OF FIRM CAPITAL GOODS: FIXING THE ERROR IN *MODEL1.4CAPITALQ*

ANONYMOUS

1. Model's Inputs

We have the value $K_{f,t}$ of the productive capital of the firm f at time t. We subdivide it in n constituent parts expressed as quotas $q_{f,i}$ with $\sum_{i=1}^{n} q_{f,i} = 1$. Quotas are specific of the "sectorial class" of the firm f.

We calculate—as sum of production costs—the prices of the n types of capital productive goods, as p_i.

With

$$k_{f,i} = q_{f,i} K_{f,0} \tag{1}$$

as investment components of the firm f in value, we obtain

$$\hat{k}_{f,i} = \frac{k_{f,i}}{p_i} \tag{2}$$

as investment components of the firm f in quantity.

$\hat{K_{f,0}}$ follows as:

$$\hat{K_{f,0}} = \sum_{i=1}^{n} \hat{k}_{f,i} \tag{3}$$

$\hat{K_{f,t}}$ is the productive capital of the firm f at time t, in quantity; i.e., *a bit strange sum of heterogeneous addenda.*

Date: October 21, 2024.

2. Dynamic

Adding the time dynamic, we have $\hat{K_{f,0}}$, obtained eliminating prices in $K_{f,0}$, becoming $\hat{K_{f,t}}$ at time t. To obtain $K_{f,t}$ we cannot use $q_{f,i}$ quotas, but $\hat{q}_{f,i}$ ones; with:

$$\hat{q}_{f,i} = \frac{\hat{k}_{f,i}}{\hat{K_{f,0}}} \tag{4}$$

We obtain:

$$K_{f,t} = \hat{K_{f,t}} \sum_{i=1}^{n} \hat{q}_{f,i} p_i \tag{5}$$

3. An example

With $K_{f,0} = 1$ and $q_f = [\frac{1}{2}, \frac{1}{2}]$, $p = [1, 2]$:

- From eq. (1) follows $k_f = [\frac{1}{2}1, \frac{1}{2}1]$;
- From eq. (2) $\hat{k}_f = [\frac{1}{2}/1, \frac{1}{2}/2] = [\frac{1}{2} + \frac{1}{4}]$;
- From eq. (3) $\hat{K_{f,0}} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$;
- From eq. (4) $\hat{q}_f = [\frac{1}{2}/\frac{3}{4}, \frac{1}{4}/\frac{3}{4}] = [\frac{2}{3}, \frac{1}{3}]$;
- From eq. (correctCalculation) $K_{f,t} = \frac{3}{4}[\frac{2}{3}1 + \frac{1}{3}2] = \frac{3}{4}\frac{4}{3} = 1$.

Using instead the wrong calculation way:

$$K_{f,t} = \hat{K_{f,t}} \sum_{i=1}^{n} q_{f,i} p_i \tag{6}$$

we had obtained: $K_{f,t} = \frac{3}{4}(\frac{1}{2}1 + \frac{1}{2}2) = \frac{3}{4}\frac{3}{2} = \frac{9}{8} = 1.125$.

4. The code

In `model1.4CapitalQ` we were implicitly using the miscalculation of eq. (6) at the end of:

```
def produce(self,model)->tuple:
```

with:

```
self.desiredCapitalSubstitutions.append(desiredCapitalQsubstitutions*\
        investmentComposition[self.sectorialClass][i]* model.investmentGoodPrices[i])
self.requiredCapitalIncrement.append(requiredCapitalQincrement*\
        investmentComposition[self.sectorialClass][i]* model.investmentGoodPrices[i])
```

being `investmentComposition` the initial quotas $q_{f,i}$.
Simplifying the eq. (5) for easier calculations we observe:

- using eq. (4) in (5) we obtain:

$$K_{f,t} = \hat{K_{f,t}} \sum_{i=1}^{n} \frac{\hat{k}_{f,i}}{\hat{K_{f,0}}} p_i = \frac{\hat{K_{f,t}}}{\hat{K_{f,0}}} \sum_{i=1}^{n} \hat{k}_{f,i} p_i \tag{7}$$

- using 2 and then 1 definitions, we have:

$$K_{f,t} = \frac{\hat{K_{f,t}}}{\hat{K_{f,0}}} \sum_{i=1}^{n} \frac{k_{f,i}}{p_i} p_i = \frac{\hat{K_{f,t}}}{\hat{K_{f,0}}} K_{f,0} \tag{8}$$

Into the code we have to calculate the sum of `desiredCapitalQsubstitutions` + the sum `requiredCapitalQincrement` and report the result to the initial capital expressed in quantity, e.g., `self.capitalQ0` set in `def settingCapitalQ` and never modified and apply the correction to obtain the new values of `desiredCapitalSubstitutions` and `requiredCapitalIncrement`, but this is only an apparent simplification, due to the presence of a lot of intermediate use of measure of the capital.as values or as quantities.

The actual simpler solution is to memorize the quotas of eq. (4) within each firm and use them in the rows of code reported above, becoming:

```
self.desiredCapitalSubstitutions.append(desiredCapitalQsubstitutions*\
         self.investmentCompositionQ[i]* model.investmentGoodPrices[i]
self.requiredCapitalIncrement.append(requiredCapitalQincrement*\
        self.investmentCompositionQ[i]* model.investmentGoodPrices[i])
```

with the novelty of `investmentCompositionQ[i]`.

Using eqs. (4) and (2) we add the following code at the end of `def settingCapitalQ`.

```
self.investmentCompositionQ=[]
for i in range(len(params['investmentGoods'])):
      self.investmentCompositionQ.append(((investmentComposition[self.sectorialClass][i]*self.capital)/ \
             investmentGoodPrices[i])/self.capitalQ)
```