

『CGEモデルにおけるベンチマーク価格の規準化方法』

武田史郎

関東学園大学経済学部
373-8515 群馬県太田市藤阿久町 200

2009/12/01

概要

CGEモデルにおけるベンチマーク価格の規準化について.

1 説明

CGE (computable general equilibrium) 分析ではベンチマークデータのもとで均衡が満たされているという前提を置き、分析がおこなわれる。均衡が成立しているということは、ベンチマークの価格、数量の下で利潤最大化条件、効用最大化条件、市場均衡条件等が満たされているということ。

ベンチマークデータは金額のデータとして与えられる。価格、数量に分割されたデータではない。これは以下の理由による。

- 投入、産出等のベンチマークデータには通常産業連関表が利用されるが、一部を除き価格、数量に分割された形にはなっていない。
- そもそもデータにおいて価格、数量に分割することが意味を持たないことも多い。CGE分析では部門・財は多くてもせいぜい100部門・財程度(通常はもっと少ない)。つまり、部門・財はかなり統合されたものを前提とする。そのように統合の程度が大きい場合には、様々な財が混ざってしまっているため数量を考える意味合いが小さい。
- サービス産業に関してはさらに数量のデータを出すことは意味が小さい。

以上のような理由から産業連関表は金額のデータとして提供されている。しかし、CGEモデルでは当然価格・数量を分割して扱うので、CGEモデルのベンチマークデータとして利用しようとするなら、たとえデータ上、価格・数量に分割するのが意味がないとしても、なんらかの方法で両者を分割する必要がある。普通は、

- ベンチマークの価格を仮定する。
- 金額データを価格で割ることで数量を求める。

という方法が採用される¹。

例えば、 V_i^0 をベンチマークにおける部門 i の生産額のデータとしたとき、ベンチマークの価格 \bar{p}_i^0 を仮定し、 $\bar{q}_i^0 = V_i^0 / \bar{p}_i^0$ でベンチマークの生産量を導出するということである。問題はベンチマークの価格をどう仮定するかということ。普通は「ベンチマークの価格=1と仮定」する。つまり、ベンチマークの金額=数量とみなす。

次の疑問が生じる。

¹逆でもよいが。

- 1 と仮定することにシミュレーションの結果は依存するか？ 1 以外の値を仮定した場合に、1 と仮定した場合と結果が変わってくるのか？

結論を先に言うと、

- シミュレーション結果は価格の規準化には依存しない。ただし、ここでのシミュレーション結果とは内生変数 (の実質値) の変化率のこと。
- 変数の絶対的な水準は規準化の方法に依存する。
- 変化率に限れば規準化の方法に依存しないので、どのような価格を仮定してもよい。
- どのような価格を仮定してもよいなら、一番単純に 1 を仮定しておくのが便利。

1.1 例

表 1: SAM (金額表示)

	部門			所得	行和	
	q_x	q_y	u	h		
市場	p_x	V_x^o	$-V_{xy}^i$	$-V_x^c$		0
	p_y	$-V_{yx}^i$	V_y^o	$-V_y^c$		0
	r	$-V_x^k$	$-V_y^k$		V^k	0
	w	$-V_x^l$	$-V_y^l$		V^l	0
	p_u			V^u	$-V^h$	0
税金	T_k	$-V_x^{tk}$			V^t	0
列和	0	0	0	0		

表の見方.

- 横に見ると市場 (財) と税, 縦に見ると部門, 所得を表す. プラスは産出、供給、マイナスは投入、購入、需要を表す。
- q_x の列は x 財部門のデータ. V_x^o は x 財の生産額、 V_{yx}^i は x 財部門の y 財の投入額、 V_x^k は資本の投入額、 V_x^l は労働の投入額である. さらに、 x 財部門の資本の投入に対しては税が課されており、 V_x^{tk} が資本に対する税の支払い額を表している。
- 同様に、 V_y^o は y 財の生産額、 V_{xy}^i は y 財部門の x 財の投入額、 V_y^k は資本の投入額、 V_y^l は労働の投入額である²。
- u の列は消費 (効用) を表している. 消費活動は、消費財を投入することで効用を生産する活動とみなすことができるので、 x 財部門、 y 財部門と同じような扱いをしている. V_x^c が x 財の消費額、 V_y^c が y 財の消費額、 V^u が消費によって生みだされた効用の価値を表している。

² y 財部門には税金は課されていないと仮定されている。

- 最後の h の列は代表的家計の所得を表している。 V^k が資本所得、 V^l が労働所得、 V^T が税収である³。代表的家計は得た所得を効用を購入することに費す。その購入額が V^h で表されている。
- 行は市場を表すが、需要額、供給額は等しくなっていないといけないので、行和はゼロとなる。
- また、 T_k の行は、産業 x の資本投入への税金を表すが、その税金は家計に全額わたるので、やはり行和はゼロとなる。
- 同様に、列は部門を表すが、収穫一定の技術の下で行動する完全競争部門を考えるので、収入＝費用 (完全分配) が成り立っていないといけない。よって、列和もゼロとなる。
- また、 h の列についても家計は得た収入は総べて消費にまわすと仮定するので列和はゼロである。

次に、表 1 を価格と数量に分割した形で表す。－付きの変数はベンチマークにおける価格、数量を意味する。

表 2: SAM (価格と数量に分解したもの)

	q_x	q_y	u	h	行和
p_x	$\bar{p}_x \bar{q}_x$	$-\bar{p}_x \bar{i}nt_{xy}$	$-\bar{p}_x \bar{c}_x$		0
p_y	$-\bar{p}_y \bar{i}nt_{yx}$	$\bar{p}_y \bar{q}_y$	$-\bar{p}_y \bar{c}_y$		0
r	$-\bar{r} \bar{k}_x$	$-\bar{r} \bar{k}_y$		$\bar{r} \bar{k}$	0
w	$-\bar{w} \bar{l}_x$	$-\bar{w} \bar{l}_y$		$\bar{w} \bar{l}$	0
p_u			$\bar{p}_u \bar{u}$	$-V^h$	0
T_k	$-t^K \bar{r} \bar{k}_x$			V^t	0
列和	0	0	0	0	

表 3: SAM の数値例 (金額表示)

	q_x	q_y	u	h	行和
p_x	140	-40	-100		0
p_y	-20	240	-220		0
r	-40	-120		160	0
w	-60	-80		140	0
p_u			320	-320	0
T_k	-20			20	0
列和	0	0	0	0	

³このモデルでは政府を明示的には扱っていない。よって、税収はそのまま代表的家計に一括の形でトランスファーされると仮定している。

2 GAMS のプログラム

表 3 の SAM を数値例とし、

- ベンチマークの価格を 1 と規準化するケース
- ベンチマークの価格を 1 以外に規準化するケース

の二つのケースを実際に CGE モデルで解き、分析結果が変わらないことを確認する。以下は、そのための GAMS のプログラム。

モデルの概略

- 二部門・二財 (x 財部門と y 財部門).
- 完全競争
- 収穫一定。生産関数は CES 型関数.
- 生産への投入物は中間財と資本と労働.
- x 部門での資本の投入には税がかかる.
- 需要側は代表的家計で表現.
- 効用関数は二つの財を消費する形の CES 型関数.
- 家計は資本、労働を生産に提供し所得を得る.
- 政府は考慮せず。税金は直接家計に一括で還流.
- 閉鎖経済.

計算結果の `report_result` の値を比較する。

```
$title Normalization of benchmark prices:
display "com: Normalization of benchmark prices:";
$ontext
Time-stamp:      <2009-12-01 20:30:16 Shiro Takeda>
$Id$
Author:          Shiro Takeda <shiro.takeda@gmail.com>
First-written:   <2008/10/22>
```

```
This is a program to show that results of the simulation (% changes in
variables) do not depend on the way of normalization of benchmark prices.
```

```
[Two cases]
```

```
Case 1: All benchmark prices are normalized to unity.
```

```
Case 2: Benchmark prices are not normalized to unity:
```

```
$offtext
```

```

* -----
* Parameters for benchmark data:
display "com: Parameters for benchmark data:";

* SAM data:
table sam "Sample SAM (Social Accounting Matrix)"
      x      y      u      h
px    140    -40    -100
py    -20    240    -220
r     -40    -120             160
w     -60    -80             140
pu                    320    -320
tk     -20             20
;
display sam;

* Parameters for benchmark value:
parameter
  v_o_x0    Benchmark value of output of x
  v_o_y0    Benchmark value of output of y
  v_i_xy0   Benchmark value of input of x in sector y
  v_i_yx0   Benchmark value of input of y in sector x
  v_k_x0    Benchmark value of input of capital in sector x
  v_k_y0    Benchmark value of input of capital in sector y
  v_l_x0    Benchmark value of input of labor in sector x
  v_l_y0    Benchmark value of input of labor in sector y
  v_c_x0    Benchmark value of consumption of x
  v_c_y0    Benchmark value of consumption of y
  v_u0      Benchmark value of utility
  v_k0      Benchmark value of capital endowment
  v_l0      Benchmark value of labor endowment
  v_h0      Benchmark value of household income
  v_tk_x0   Benchmark value of tax on capital in sector x
  v_t0      Benchmark value of tax revenue
;
v_o_x0 = sam("px","x");
v_o_y0 = sam("py","y");
v_i_xy0 = - sam("px","y");
v_i_yx0 = - sam("py","x");
v_k_x0 = - sam("r","x");
v_k_y0 = - sam("r","y");
v_l_x0 = - sam("w","x");
v_l_y0 = - sam("w","y");
v_c_x0 = - sam("px","u");
v_c_y0 = - sam("py","u");
v_tk_x0 = - sam("tk","x");

v_u0 = v_c_x0 + v_c_y0;
v_k0 = v_k_x0 + v_k_y0;
v_l0 = v_l_x0 + v_l_y0;
v_h0 = v_u0;

display v_o_x0, v_o_y0, v_i_xy0, v_i_yx0, v_k_x0, v_k_y0, v_l_x0, v_l_y0,
       v_c_x0, v_c_y0, v_u0, v_k0, v_l0, v_h0;

* Parameters for benchmark price:
parameter
  p_x0      Benchmark price of x

```

```

    p_y0      Benchmark price of y
    r0        Benchmark price of capital
    w0        Benchmark price of labor
    p_u0      Benchmark price of utility
;
parameter
    tk0       Benchmark tax rate on capital in sector x;

*           Parameters for benchmark quantity:
parameter
    q_x0      Benchmark output of x
    q_y0      Benchmark output of y
    int_yx0   Benchmark input of y in sector x
    int_xy0   Benchmark input of x in sector y
    k_x0      Benchmark input of k in sector x
    k_y0      Benchmark input of k in sector y
    l_x0      Benchmark input of l in sector x
    l_y0      Benchmark input of l in sector y
    c_x0      Benchmark consumption of x
    c_y0      Benchmark consumption of y
    u0        Benchmark utility

    k0        Benchmark capital endowment
    l0        Benchmark labor endowment
;

*           EOS parameters in production and utility functions:
parameter
    sig_va    EOS: capital vs labor in production
    sig_u     EOS: consumption of x and y in utility
;
sig_va = 0.5;
sig_u = 1;

*           Tax rate and other parameters:
parameter
    tk        Tax rate on capital in sector x
    scale_k   Scale factor for capital
    scale_l   Scale factor for labor
;
scale_k = 1;
scale_l = 1;
display scale_k, scale_l;

*           -----
*           Specification of functional forms:
display "com: Specification of functional forms:";
$ontext

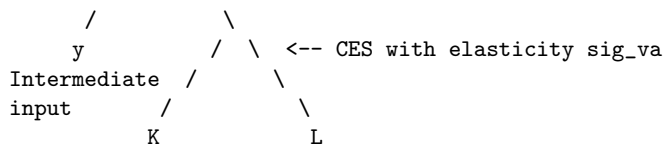
Production functions:

+ Production function of sector x:

    Two stage CES: the first stage is Leontief type and the second stage is CES.

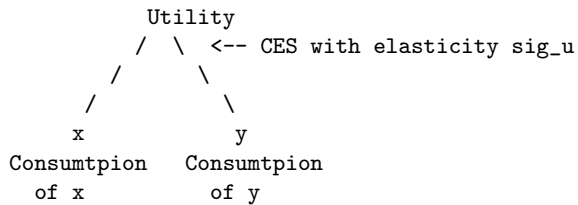
        Output of sector x
        / \ <-- Leontief
        /   \

```



The same type of production function is applied to sector y.

+ Utility function:



\$offtext

```

* -----
*      Model by MPSGE:
display "com: Model by MPSGE:";

```

\$ontext

\$model:price_norm

\$sectors:

```

q_x      ! Output of x
q_y      ! Output of y
u        ! Utility

```

\$commodities:

```

p_x      ! Price of x
p_y      ! Price of y
r        ! Price of capital
w        ! Price of labor
p_u      ! Price of utility

```

\$consumer:

```

h        ! Income of a representative household

```

* Production function of sector x:

```

$prod:q_x      s:0 va:sig_va
o:p_x          q:q_x0          p:p_x0
i:p_y          q:int_yx0       p:p_y0
i:r            q:k_x0          p:((1+tk0)*r0) a:h t:tk va:
i:w            q:l_x0          p:w0          va:

```

* Production function of sector y:

```

$prod:q_y      s:0 va:sig_va
o:p_y          q:q_y0          p:p_y0
i:p_x          q:int_xy0       p:p_x0
i:r            q:k_y0          p:r0 va:
i:w            q:l_y0          p:w0 va:

```

* Utility function:

```

$prod:u      s:sig_u
  o:p_u      q:u0      p:p_u0
  i:p_x      q:c_x0     p:p_x0
  i:p_y      q:c_y0     p:p_y0

*      Income:
$demand:h
  d:p_u      q:u0
  e:r        q:(scale_k*k0)
  e:w        q:(scale_l*l0)

$offtext

$sysinclude mpsgeset price_norm

parameter
  report_result      Parameter for reporting results (check that % changes in variables are the same in
;

*
*      -----
*      Case 1: all benchmark prices are normalized to unity:
display "com: Case 1: all benchmark prices are normalized to unity:";

*      All benchmark prices are normalized to unity:
p_x0 = 1;
p_y0 = 1;
r0 = 1;
w0 = 1;
p_u0 = 1;
display p_x0, p_y0, r0, w0, p_u0;

*      Derivation of benchmark quantity:
q_x0 = v_o_x0 / p_x0;
q_y0 = v_o_y0 / p_y0;
int_yx0 = v_i_yx0 / p_y0;
int_xy0 = v_i_xy0 / p_x0;
k_x0 = v_k_x0 / r0;
k_y0 = v_k_y0 / r0;
l_x0 = v_l_x0 / w0;
l_y0 = v_l_y0 / w0;
c_x0 = v_c_x0 / p_x0;
c_y0 = v_c_y0 / p_y0;
u0 = v_u0 / p_u0;
k0 = v_k0 / r0;
l0 = v_l0 / w0;
display q_x0, q_y0, int_yx0, int_xy0, k_x0, k_y0, l_x0, l_y0, c_x0, c_y0, u0, k0, l0;

tk0 = v_tk_x0 / (r0 * k_x0);
display tk0;
tk = tk0;

*
*      -----
*      Benchmark replication.
display "com: Benchmark replication.";

*      Initialization of endogenous variables:
q_x.l = 1;
q_y.l = 1;

```



```

u.l = 1;
p_x.l = p_x0;
p_y.l = p_y0;
r.l = r0;
w.l = w0;
p_u.l = p_u0;
h.l = v_h0;

price_norm.iterlim = 0;
$include price_norm.gen
solve price_norm using mcp;

* -----
* Cleanup calculation.
display "com: Cleanup calculation.";
price_norm.iterlim = 10000;
$include price_norm.gen
solve price_norm using mcp;

* -----
* Counter-factual experiment: a rise in labor endowment
display "com: Counter-factual experiment: a rise in labor endowment";

scale_k = 1.03;
$include price_norm.gen
solve price_norm using mcp;
scale_k = 1.0;

report_result("q_x","Case1") = 100 * (q_x.l - 1);
report_result("q_y","Case1") = 100 * (q_y.l - 1);
report_result("u","Case1") = 100 * (u.l - 1);
report_result("p_x","Case1") = 100 * ((p_x.l/p_u.l)/(p_x0/p_u0) - 1);
report_result("p_y","Case1") = 100 * ((p_y.l/p_u.l)/(p_y0/p_u0) - 1);
report_result("w","Case1") = 100 * ((w.l/p_u.l)/(w0/p_u0) - 1);
report_result("r","Case1") = 100 * ((r.l/p_u.l)/(r0/p_u0) - 1);
report_result("h","Case1") = 100 * ((h.l/p_u.l)/(v_h0/p_u0) - 1);

* -----
* Case 2: benchmark prices are not normalized to unity:
display "com: Case 2: benchmark prices are not normalized to unity:";

* Normalization of benchmark prices (not equal to unity):
p_x0 = 2;
p_y0 = 3;
r0 = 4;
w0 = 0.5;
p_u0 = 1;
display p_x0, p_y0, r0, w0, p_u0;

* Derivation of benchmark quantity:
q_x0 = v_o_x0 / p_x0;
q_y0 = v_o_y0 / p_y0;
int_yx0 = v_i_yx0 / p_y0;
int_xy0 = v_i_xy0 / p_x0;
k_x0 = v_k_x0 / r0;
k_y0 = v_k_y0 / r0;
l_x0 = v_l_x0 / w0;
l_y0 = v_l_y0 / w0;

```

```

c_x0 = v_c_x0 / p_x0;
c_y0 = v_c_y0 / p_y0;
u0 = v_u0 / p_u0;
k0 = v_k0 / r0;
l0 = v_l0 / w0;
display q_x0, q_y0, int_yx0, int_xy0, k_x0, k_y0, l_x0, l_y0, c_x0, c_y0, u0, k0, l0;

tk0 = v_tk_x0 / (r0 * k_x0);
display tk0;
tk = tk0;

* -----
* Benchmark replication.
display "com: Benchmark replication.";

* Initialization of endogenous variables:
q_x.l = 1;
q_y.l = 1;
u.l = 1;
p_x.l = p_x0;
p_y.l = p_y0;
r.l = r0;
w.l = w0;
p_u.l = p_u0;
h.l = v_h0;

price_norm.iterlim = 0;
$include price_norm.gen
solve price_norm using mcp;

* -----
* Cleanup calculation.
display "com: Cleanup calculation.";
price_norm.iterlim = 10000;
$include price_norm.gen
solve price_norm using mcp;

* -----
* Counter-factual experiment: a rise in labor endowment
display "com: Counter-factual experiment: a rise in labor endowment";

scale_k = 1.03;
$include price_norm.gen
solve price_norm using mcp;
scale_k = 1.0;

report_result("q_x", "Case2") = 100 * (q_x.l - 1);
report_result("q_y", "Case2") = 100 * (q_y.l - 1);
report_result("u", "Case2") = 100 * (u.l - 1);
report_result("p_x", "Case2") = 100 * ((p_x.l/p_u.l)/(p_x0/p_u0) - 1);
report_result("p_y", "Case2") = 100 * ((p_y.l/p_u.l)/(p_y0/p_u0) - 1);
report_result("w", "Case2") = 100 * ((w.l/p_u.l)/(w0/p_u0) - 1);
report_result("r", "Case2") = 100 * ((r.l/p_u.l)/(r0/p_u0) - 1);
report_result("h", "Case2") = 100 * ((h.l/p_u.l)/(v_h0/p_u0) - 1);

* -----
* Compare results in two cases:
display "com: Compare results in two cases:";

```

```
$ontext
```

```
Check that results from two experiments are identical.
```

```
$offtext
```

```
display report_result;
```

```
* -----
```

```
* local Variables:
```

```
* mode: gams
```

```
* fill-column: 80
```

```
* End:
```