応用一般均衡分析入門
第 5 章：一般均衡モデルの解き方 *

武田 史郎 †

Date: 2023/05/08,
Version 4.0

目次

1 導入 · · · · · 1
2 一般均衡モデル · · · · · 2
2.1 伝統的アプローチによる表現 · · · · · 2
2.2 双対アプローチによる均衡条件 · · · · · 3
3 GAMS のプログラム · · · · · 4
3.1 記号の説明 · · · · · 4
3.2 プログラムの解説 · · · · · 4
3.3 計算結果 · · · · · 15
4 終わりに · · · · · 18
参考文献 · · · · · 18
5 履歴 · · · · · 18

1 導入

今回の内容について

- 第 5 章では第 2 章で説明した一般均衡モデルを GAMS で解く方法の解説をおこなう。
- モデルを GAMS で解くということは結局そのモデルを GAMS のプログラムとしてどのように表現するかということである。第 2 章のモデルを第 4 章で説明した MCP として記述し、解く方法について説明する。

[注]

この章では一般均衡モデルを数値的に解くという作業をおこなうが、それは CGE モデルを解くこととは少し異なっている。というのは、CGE モデルでは最初の段階の均衡（基準均衡）は既知

---

*このファイルの配布場所: https://shirotakeda.github.io/ja/research-ja/cge-howto.html

†所属：京都産業大学経済学部. Website: https://shirotakeda.github.io/ja/

という前提、もう少し具体的には、与えられた基準データが初期の均衡になっているという前提を置くのに対し、ここでは未知の均衡を求めるという形でモデルを解くからである。CGE モデルを解くというケースを説明するのは、パラメータのカリブレーションを説明する第 8 章以降となる。

モデルを解くにあたって、財、生産要素を以下のように特定化する。

- 財は AGR、MAN、SER の 3 財（$i = \{\mathrm{AGR}, \mathrm{MAN}, \mathrm{SER}\}$）とする。一つの財は一つの部門によって生産されるので部門も 3 つとなる。
- 生産要素は労働と資本の 2 つ（$f = \{\mathrm{LAB}, \mathrm{CAP}\}$）とする。

さらにモデルに次の 2 つの税金を導入する。

- 消費に対する従価税
- 生産要素の投入に対する従価税（シミューレションでは労動投入への税）

財 i に対する消費税の従価税率を t_i^C とする。消費者が直面する価格は $(1+t_i^C)p_i$ となる。一方、部門 i が投入する生産要素 f に対する従価税率を t_{fi}^F とする。企業が直面する生産要素 f の価格は $(1+t_{fi}^F)p_f^F$ となる。ここでは政府を明示的には扱わないので、どちらの税収もそのまま家計に還元される（家計に一括で還元される）と仮定する。

2　一般均衡モデル

2.1　伝統的アプローチによる表現

以下は伝統的アプローチによる均衡条件の表現である。モデルについて詳しくは第 2 章を参照して欲しい。

$$y_i = \left[\sum_j \alpha_{ji}^x (x_{ji})^{\frac{\sigma_i-1}{\sigma_i}} + \alpha_i^v (v_i^a)^{\frac{\sigma_i-1}{\sigma_i}}\right]^{\frac{\sigma_i}{\sigma_i-1}} \qquad \{y_i\}_{i=1,\cdots,n} \tag{1}$$

$$p_i y_i^{\frac{1}{\sigma_i}} \alpha_{ji}^x (x_{ji})^{-\frac{1}{\sigma_i}} - p_j = 0 \qquad \{x_{ij}\}_{i,j=1,\cdots,n} \tag{2}$$

$$p_i y_i^{\frac{1}{\sigma_i}} \alpha_i^v (v_i^a)^{-\frac{1}{\sigma_i}} - p_i^{va} = 0 \qquad \{v_i^a\}_{i=1,\cdots,n} \tag{3}$$

$$a_{ij}^x = \frac{x_{ij}}{y_j} \qquad \{a_{ij}^x\}_{i,j=1,\cdots,n} \tag{4}$$

$$a_i^v = \frac{v_i^a}{y_i} \qquad \{a_i^v\}_{i=1,\cdots,n} \tag{5}$$

$$p_i^{va} (v_i^a)^{\frac{1}{\sigma_i^v}} \beta_{fi}^v (v_{fi})^{-\frac{1}{\sigma_i^v}} - (1+t_{fi}^F) p_f^F = 0 \qquad \{v_{fi}\}_{i=1,\cdots,n,f=1,\cdots,k} \tag{6}$$

$$a_{fi}^F = \frac{v_{fi}}{v_i^a} \qquad \{a_{fi}^F\}_{f=1,\cdots,k,i=1,\cdots,n} \tag{7}$$

$$d_i = \left[\frac{\gamma_i}{(1+t_i^C)p_i}\right]^{\sigma^c} \frac{m}{\sum_j (\gamma_j)^{\sigma^c} \left[(1+t_j^C)p_j\right]^{1-\sigma^c}} \qquad \{d_i\}_{i=1,\cdots,n} \tag{8}$$

$$y_i = \sum_j x_{ij} + d_i \qquad \{p_i\}_{i=1,\cdots,n} \tag{9}$$

$$\left[\sum_f \beta^v_{fi}(v_{fi})^{\frac{\sigma^v_i - 1}{\sigma^v_i}}\right]^{\frac{\sigma^v_i}{\sigma^v_i - 1}} = v^a_i \qquad \{p^{va}_i\}_{i=1,\cdots,n} \tag{10}$$

$$\bar{v}_f = \sum_j v_{fj} \qquad \{p^F_f\}_{f=1,\cdots,k} \tag{11}$$

$$m = \sum_f p^F_f \bar{v}_f + \sum_i t^C_i p_i d_i + \sum_{i,f} t^F_{fi} p^F_f v_{fi} \qquad \{m\} \tag{12}$$

$$u = \left[\sum_i \gamma_i (d_i)^{\frac{\sigma^C - 1}{\sigma^C}}\right]^{\frac{\sigma^C}{\sigma^C - 1}} \qquad \{u\} \tag{13}$$

税が導入された部分については税込みの価格に修正し、(12) 式では税収を家計の所得に加えている。また、双対アプローチのモデルに現れる変数と共通化するために幾つか変数を追加している。

2.2 双対アプローチによる均衡条件

次に双対アプローチによる均衡条件の表現である。

$$\left[\sum_j (\alpha^x_{ji})^{\sigma_i}(p_j)^{1-\sigma_i} + (\alpha^v_i)^{\sigma_i}(p^{va}_i)^{1-\sigma_i}\right]^{\frac{1}{1-\sigma_i}} - p_i = 0 \qquad \{y_i\}_{i=1,\cdots,n} \tag{14}$$

$$\left[\sum_f (\beta^v_{fi})^{\sigma^v_i}\left[(1+t^F_{fi})p^F_f\right]^{1-\sigma^v_i}\right]^{\frac{1}{1-\sigma^v_i}} - p^{va}_i = 0 \qquad \{v^a_i\}_{i=1,\cdots,n} \tag{15}$$

$$a^x_{ji} = \left[\frac{\alpha^x_{ji}}{p_j}\right]^{\sigma_i}\left[\sum_l (\alpha^x_{li})^{\sigma_i}(p_l)^{1-\sigma_i} + (\alpha^v_i)^{\sigma_i}(p^{va}_i)^{1-\sigma_i}\right]^{\frac{\sigma_i}{1-\sigma_i}} \qquad \{a^x_{ji}\}_{i,j=1,\cdots,n} \tag{16}$$

$$a^v_i = \left[\frac{\alpha^v_i}{p^{va}_i}\right]^{\sigma_i}\left[\sum_l (\alpha^x_{li})^{\sigma_i}(p_l)^{1-\sigma_i} + (\alpha^v_i)^{\sigma_i}(p^{va}_i)^{1-\sigma_i}\right]^{\frac{\sigma_i}{1-\sigma_i}} \qquad \{a^v_i\}_{i=1,\cdots,n} \tag{17}$$

$$x_{ij} = a^x_{ij} y_j \qquad \{x_{ij}\}_{i,j=1,\cdots,n} \tag{18}$$

$$a^F_{fi} = \left[\frac{\beta^v_{fi}}{(1+t^F_{fi})p^F_f}\right]^{\sigma^v_i}\left[\sum_k (\beta^v_{ki})^{\sigma^v_i}\left[(1+t^F_{ki})p^F_k\right]^{1-\sigma^v_i}\right]^{\frac{\sigma^v_i}{1-\sigma^v_i}} \qquad \{a^F_{fi}\}_{i=1,\cdots,n,f=1,\cdots,k} \tag{19}$$

$$v_{fi} = a^F_{fi} v^a_i \qquad \{v_{fi}\}_{f=1,\cdots,k,i=1,\cdots,n} \tag{20}$$

$$d_i = \left[\frac{\gamma_i}{(1+t^C_i)p_i}\right]^{\sigma^c}\left[\sum_j (\gamma_j)^{\sigma^c}\left[(1+t^C_j)p_j\right]^{1-\sigma^c}\right]^{\frac{\sigma^c}{1-\sigma^c}} u \qquad \{d_i\}_{i=1,\cdots,n} \tag{21}$$

$$y_i = \sum_j a^x_{ij} y_j + d_i \qquad \{p_i\}_{i=1,\cdots,n} \tag{22}$$

$$v^a_i = a^v_i y_i \qquad \{p^{va}_i\}_{i=1,\cdots,n} \tag{23}$$

$$\bar{v}_f = \sum_i a^F_{fi} v^a_i \qquad \{p^F_f\}_{f=1,\cdots,k} \tag{24}$$

$$m = \sum_f p_f^F \bar{v}_f + \sum_i t_i^C p_i d_i + \sum_{i,f} t_{fi}^F p_f^F v_{fi} \qquad \{m\} \tag{25}$$

$$u \left[\sum_i (\gamma_i)^{\sigma^c} \left[(1+t_i^C) p_i \right]^{1-\sigma^c} \right]^{\frac{1}{1-\sigma^c}} = m \qquad \{u\} \tag{26}$$

こちらについても、税金を導入したことによる修正を加え、変数を追加している。

3 GAMS のプログラム

以下では、第 2 節の一般均衡モデルを GAMS で解くプログラムを説明していく。モデルの表現として**「伝統的アプローチ」**と**「双対アプローチ」**の 2 つを使ったが、以下では基本的に双対アプローチによる表現の方を利用していく。もちろんどちらのアプローチを利用しても同じモデルであるので求められる均衡は同じである。双対アプローチの記述を利用して書いたプログラムは ge_sample_dual.gms であり、これを説明していく。同じモデルを伝統的アプローチで記述したものは ge_sample_trad.gms である。後で両者を比較し、同じ結果が求められるかどうか確認して欲しい。

以下でプログラムの説明をしていくが、自分でプログラムを書き実行しながら解説を読んでいくようにして欲しい。

3.1 記号の説明

表 1〜表 3 がモデルに現われる変数、記号の説明である。

表 1：集合の説明

モデル	説明	プログラム	中身
i, j	財の集合	i, j	AGR, MAN, SER
f	生産要素の集合	f	LAB, CAP

3.2 プログラムの解説

それでは以下でプログラムを説明していく。

```
*       ------------------------------------------------------------
*       集合の宣言
set     i 財の集合                / agr, man, ser /
        f 生産要素の集合          / lab, cap /
;
*       alias の作成
alias(i,j), (i,ii), (f,ff);
display i, j, f;
```

表 2：変数の説明

モデル	説明	プログラム
y_i	生産量	y(i)
v_i^a	合成生産要素	v_a(i)
a_i^v	単位合成生産要素需要	a_v(i)
x_{ij}	中間投入需要	x(i,j)
a_{ji}^x	単位中間投入需要	a_x(j,i)
v_{fi}	生産要素需要	vf(f,i)
a_{fi}^F	単位生産要素需要	a_f(f,i)
d_i	消費需要	d(i)
p_i	財価格	p(i)
p_i^{va}	合成生産要素の価格	p_va(i)
p_f^F	生産要素の価格	p_f(f)
u	効用水準	u
m	所得	m

表 3：パラメータ、外生変数の説明

モデル	説明	プログラム
α_{ji}^x	生産関数内のウェイトパラメータ	alpha_x(j,i)
α_i^v	生産関数内のウェイトパラメータ	alpha_v(i)
β_{fi}^v	生産関数内のウェイトパラメータ	beta_v(f,i)
γ_i	効用関数内のウェイトパラメータ	gamma(i)
σ_i	中間投入と合成生産要素の間の代替の弾力性	sig(i)
σ_i^v	生産要素間の代替の弾力性	sig_v(i)
σ^c	消費財間の代替の弾力性	sig_c
$\bar{v}_f$	生産要素賦存量	v_bar(f)
t_i^C	消費税率	t_c(i)
t_{fi}^F	生産要素投入への課税率	t_f(f,i)

ここで集合の宣言・定義をしている。財を表す集合 i と、生産要素を表す f という集合を定義している。最初に説明した通り、財は 3 つ、生産要素は 2 つである。さらに、alias 命令で集合のコピーを作成している。

```
*       ------------------------------------------------------------
*       生産関数・効用関数のパラメータ
$ontext
生産関数、効用関数内のウェイトパラメータの宣言。
$offtext
parameter
    alpha_x(j,i)     生産関数のウェイトパラメータ
    alpha_v(i)       生産関数のウェイトパラメータ
    beta_v(f,i)      生産関数のウェイトパラメータ
    gamma(i)         効用関数のウェイトパラメータ
;
```

ここでは、生産関数、効用関数内のウェイトパラメータを定義している。モデル内のパラメータとプログラムのパラメータの対応は表 3 で確認して欲しい。

```
table t_alpha(*,i) "alpha_x と alpha_v の値"
                    agr           man           ser
agr         0.045918367   0.001111111          0.04
man         0.005102041   0.027777778   0.017777778
ser         0.020408163   0.017777778   0.017777778
v           0.326530612   0.444444444   0.284444444
;
```

ここではパラメータ t_alpha を table 命令によって宣言・定義している。t_alpha はパラメータ alpha_x、alpha_v の値を入れておくパラメータである。table 命令は parameter 命令と同様、パラメータを宣言・定義するための命令の一つで、特に 2 次元のパラメータを定義するときに用いられる。

table 命令の基本的な書式は次の通りである。

```
table t_name パラメータの説明
           a      b
x        100     10
y         20     50
;
```

以上のようにパラメータ t_name を定義してやると次のように定義した場合と同じ効果を持つ。

```
t_name("x","a") = 100;
t_name("y","a") = 20;
t_name("x","b") = 10;
t_name("y","b") = 50;
```

table 命令を用いるとパラメータにどのような値を代入しているかが非常にわかりやすい。そのため 2 次元の形式のパラメータを定義する際によく利用される。table 命令について詳しくは GAMS のマニュアルの “Data Entry: Parameters, Scalars and Tables” を見て欲しい。

```
table t_beta_v(f,i) beta_v の値
                 agr      man         ser
lab         0.390625     0.16    0.390625
cap         0.140625     0.36    0.140625
;
```

ここでは t_alpha と同様に t_beta を定義する。t_beta は beta_v の値を設定しているパラメータである。

```
*       t_alpha の値 → alpha_x & alpha_v に代入
alpha_x(j,i) = t_alpha(j,i);
alpha_v(i) = t_alpha("v",i);

*       alpha の値をチェック
display alpha_x, alpha_v;

*       beta についても同様
beta_v(f,i) = t_beta_v(f,i);
display beta_v;
```

ここでは t_alpha と t_beta の値を alpha_x、alpha_v、beta_v に代入している。 t_alpha と t_beta の値には根拠があるのだが、ここでは深く考えなくてよい。とりあえずウェイトパラメータは外生的に決定するパラメータであり、ここではその値を設定しているとだけ考えればよい。

```
*       gamma の値の設定
gamma("agr") = 0.037808642;
gamma("man") = 0.37345679;
gamma("ser") = 0.037808642;
```

```
display gamma;
```

ここでは gamma の値を設定している。やはり外生的に値を設定する。

```
*       ------------------------------------------------------------
*       代替の弾力性パラメータ
$ontext
+ 生産関数、効用関数の中の代替の弾力性（elasticity of substitution,
  EOS）も外生的に設定する。
+ 1 と 0（つまり、Cobb-Douglas と Leontief 型）には設定できないので注意。
$offtext
parameter
    sig(i)      生産要素とそれ以外の投入の間の EOS
    sig_v(i)    生産要素間の EOS
    sig_c       消費における EOS
;
sig(i) = 0.5;
sig_v(i) = 0.5;
sig_c = 0.5;
display sig, sig_v, sig_c;
```

ここでは代替の弾力性パラメータの設定をしている。どれも CES 関数内の代替の弾力性パラメータであるが、CES 関数の弾力性の値については通常次のようなことが成り立つ

- 代替の弾力性＝ 0 → 関数は Leontief 関数になる
- 代替の弾力性＝ 1 → 関数は Cobb-Doubles 関数になる

しかし、このプログラムでは代替の弾力性を 0 や 1 にすることはできない[1)]。というのは、「代替の弾力性を 0 や 1 にすると CES 関数が Leontief 関数や Cobb-Douglas 関数になる」というのは、厳密には**「CES 関数において代替の弾力性を 0 や 1 に近づけていくと、関数が Lentief 関数や Cobb-Douglas 関数に収束していく」**という話であるからである。CES 関数の代替の弾力性にそのまま 0 や 1 を代入した場合、分母にゼロが入る部分が出てくるので GAMS では単にエラーになってしまう。とりあえずここでは代替の弾力性を全て 0.5 と置いている。0 や 1 でなければどういう値を設定してもよい（ただし、値によっては均衡が計算できなくなるようなことはあるかもしれないが）。

```
*       ------------------------------------------------------------
*       外生変数
```

1) 厳密には ge_sample_dual.gms では代替の弾力性を 0 にしてもよい。つまり Leontief 関数を想定してもよい（その結果、正常な均衡が存在するかどうかは別の話だが）。しかし、同じモデルを伝統的アプローチで記述した ge_sample_trad.gms では 0 にすることはできない。実際代替の弾力性に 0 を設定すると「division by zero」エラーが生じるはずである。

```
$ontext
+ モデルの外生変数は生産要素の賦存量。
$offtext
parameter
    v_bar(f)     生産要素の賦存量（外生的）
    v_bar0(f)    生産要素の賦存量（外生的）;
v_bar(f) = 180;
v_bar0(f) = v_bar(f);
display v_bar;
```

ここではモデルの外生変数を設定している。まずは生産要素の賦存量の設定である。労働、資本のどちらも初期賦存量を 180 と仮定している。後に、生産要素賦存量を変更するというシミュレーションをおこなうため、賦存量を表すパラメータ v_bar と、その初期値を保持するためのパラメータ v_bar0 の 2 つを導入し、区別している。

```
parameter
    t_c(i)       消費に対する従価税率
    t_f(f,i)     生産要素の投入に対する従価税率
;
*       税率の初期値はゼロとする。
t_c(i) = 0;
t_f(f,i) = 0;
```

ここでは税率を表すパラメータを宣言している。t_c(i) が財 i に対する消費税率、t_f(f,i) が部門 i における生産要素 f の投入に対する税率である。後のシミュレーションでは生産要素への課税として、労働課税しか考えない。よって、t_f("cap",i) は常にゼロということになる。どちらの税金も最初はゼロ、つまり税率の初期値は 0 と仮定している。

```
*       ----------------------------------------------------------------
*       変数の宣言
variables
    y(i)         生産量
    v_a(i)       合成生産要素
    a_x(j,i)     単位投入需要
    x(j,i)       投入需要
    a_v(i)       単位合成生産要素需要
    a_f(f,i)     単位生産要素需要
    vf(f,i)      生産要素需要
    d(i)         消費需要
    p(i)         財の価格
    p_va(i)      合成生産要素の価格
    p_f(f)       生産要素の価格
    u            効用水準
```

```
    m            所得
;
```

ここでは変数（モデルの内生変数）の宣言をしている。それぞれがモデルにおけるどの変数に対応しているかは表 2 で確認して欲しい。

```
*       --------------------------------------------------------------
*       式の宣言
equations
    e_y(i)       生産における利潤最大化条件
    e_v_a(i)     生産要素合成における利潤最大化条件
    e_a_x(j,i)   単位投入需要
    e_x(j,i)     投入需要
    e_a_v(i)     単位合成生産要素需要
    e_a_f(f,i)   単位生産要素需要
    e_vf(f,i)    生産要素需要
    e_d(i)       消費需要
    e_p(i)       財の市場均衡
    e_p_va(i)    合成生産要素の市場均衡
    e_p_f(f)     生産要素の市場均衡
    e_u          支出＝所得
    e_m          所得の定義式
;
```

ここでは式を宣言している。

```
*       --------------------------------------------------------------
*       式の定義
$ontext
式の定義については解説書の方を参照。
$offtext

*       生産における利潤最大化条件
e_y(i) .. (sum(j, alpha_x(j,i)**sig(i) * p(j)**(1-sig(i)))
            + alpha_v(i)**sig(i) * p_va(i)**(1-sig(i)))**(1/(1-sig(i)))
            - p(i) =e= 0;

*       生産要素合成における利潤最大化条件
e_v_a(i) .. (sum(f, beta_v(f,i)**sig_v(i)
              * ((1+t_f(f,i))*p_f(f))**(1-sig_v(i))))**(1/(1-sig_v(i)))
              - p_va(i) =e= 0;

*       単位投入需要
e_a_x(j,i) ..
          a_x(j,i) =e= (alpha_x(j,i)/p(j))**(sig(i))
```

```
              * (sum(ii, alpha_x(ii,i)**sig(i) * p(ii)**(1-sig(i)))
                  + alpha_v(i)**sig(i) * p_va(i)**(1-sig(i))
              )**(sig(i)/(1-sig(i)));

*       単位合成生産要素需要
e_a_v(i) .. a_v(i) =e= (alpha_v(i)/p_va(i))**(sig(i))
              * (sum(j, alpha_x(j,i)**sig(i) * p(j)**(1-sig(i)))
                  + alpha_v(i)**sig(i) * p_va(i)**(1-sig(i))
              )**(sig(i)/(1-sig(i)));

*       単位生産要素需要
e_a_f(f,i) ..
          a_f(f,i) =e=
              (beta_v(f,i)/((1+t_f(f,i))*p_f(f)))**(sig_v(i))
              * (sum(ff, beta_v(ff,i)**sig_v(i)
                  * ((1+t_f(ff,i))*p_f(ff))**(1-sig_v(i)))
              )**(sig_v(i)/(1-sig_v(i)));

*       消費需要
e_d(i) .. d(i) =e=
          u * (gamma(i)/((1+t_c(i))*p(i)))**(sig_c)
            * (sum(j, gamma(j)**(sig_c)*((1+t_c(j))*p(j))**(1-sig_c))
            )**(sig_c/(1-sig_c));

*       財の市場均衡
e_p(i) .. y(i) =e= sum(j, a_x(i,j)*y(j)) + d(i);

*       合成生産要素の市場均衡
e_p_va(i) .. v_a(i) =e= a_v(i)*y(i);

*       生産要素の市場均衡
e_p_f(f) .. v_bar(f) =e= sum(i, a_f(f,i)*v_a(i));

*       支出＝所得
e_u ..    u * (sum(j, gamma(j)**(sig_c)
            * ((1+t_c(j))*p(j))**(1-sig_c)))**(1/(1-sig_c)) =e= m;

*       所得の定義式
e_m .. m =e= sum(f, p_f(f)*v_bar(f))
            + sum(i, t_c(i)*p(i)*d(i))
            + sum((f,i), t_f(f,i)*p_f(f)*a_f(f,i)*v_a(i));

*       投入需要
e_x(j,i) .. x(j,i) =e= a_x(j,i) * y(i);

*       生産要素需要
e_vf(f,i) .. vf(f,i) =e= a_f(f,i) * v_a(i);
```

式の定義は基本的に第 2.2 節における式をそのまま GAMS の数式として表現したものである。少し煩雑なプログラムであるが、練習になるので、第 2.2 節における式を見て、自分自身で打ち込むようにして欲しい。モデル自体はそれほど複雑なモデルではなく、むしろ非常に単純なモデルである。しかし、実際にそのモデルをプログラム上で表現するとなると、意外に複雑に感じる。一つの理由はプログラムでは括弧を多用するためであろう。括弧については、その数、位置（開き括弧と閉じ括弧の対応関係）を間違えやすいので注意して欲しい。

```
*       ----------------------------------------------------------------
*       モデルの宣言
$ontext
+ MCP タイプのモデルとしてモデルを定義する。
+ 各式に対してその式に対応する変数を指定する。

$offtext
model ge_sample_dual 一般均衡モデル（双対アプローチ） /
                e_y.y, e_v_a.v_a, e_a_x.a_x, e_a_v.a_v, e_a_f.a_f,
                e_d.d, e_p.p, e_p_va.p_va, e_p_f.p_f, e_u.u, e_m.m,
                e_x.x, e_vf.vf
                /;
```

ここではモデルの定義をしている。MCP のモデルとして定義するので式に対し変数を対応させている。

```
*       ----------------------------------------------------------------
*       変数の下限値
*       数量を表す変数の下限は 0 とする。
y.lo(i) = 0;
v_a.lo(i) = 0;
a_x.lo(j,i) = 0;
a_v.lo(i) = 0;
a_f.lo(f,i) = 0;
d.lo(i) = 0;
x.lo(j,i) = 0;
vf.lo(f,i) = 0;

*       価格変数については下限を 0 ではない 0 に非常に近い値にする。
p.lo(i) = 1e-6;
p_va.lo(i) = 1e-6;
p_f.lo(f) = 1e-6;
u.lo = 0;
m.lo = 0;
```

ここでは変数の下限値を設定している。理論的には全ての変数は非負の値をとるべきであり、よってその下限値は 0 とすべきである。実際、数量を表す変数については 0 を下限値に設定している。しかし、価格を表す変数については 0 ではなく、0 に近い非常に小さい正の値、具体的には 1e-6（$= 1/10^6$）を設定している。

価格変数についてだけ 0 にしていないのは、GAMS の実行時に**「division by zero」エラーが生じることを避けるため**である。価格変数はモデルの式において分母に入ってきている部分が多い。従って、計算の過程において価格が 0 になってしまうと「division by zero」エラーが生じてしまう。下限値に 0 を設定しておくと、均衡として出てくる解では正の値になる場合でも、計算過程において 0 となってしまう可能性がある。この問題を避けるため下限値として、0 ではないが非常に小さい正の値を設定している。これは CGE モデルを解くケースだけでなく、他のケースにおいて生じうる問題である。**理論的には非負制約でよい場合（つまり、下限を 0 に設定すればよい場合）であっても、非常に小さい正の値を設定するのが望ましいことが GAMS ではよくあるので覚えておくとよい。**

```
*       --------------------------------------------------------------
*       変数の初期値
y.l(i) = 10;
v_a.l(i) = 10;
a_x.l(j,i) = 10;
a_v.l(i) = 10;
a_f.l(f,i) = 10;
d.l(i) = 10;
x.l(j,i) = a_x.l(j,i) * y.l(i);
vf.l(f,i) = a_f.l(f,i) * v_a.l(i);
p.l(i) = 1;
p_va.l(i) = 1;
p_f.l(f) = 1;
u.l = 300;
m.l = 300;
```

ここでは変数の初期値を設定している。ここで指定した値がモデルを解く際の変数の初期値として利用される。

GAMS でモデルを解く際には変数の初期値の設定が非常に重要になる。モデルが解けるかどうかが初期値の与え方に非常に強く依存している。一般に、**「変数の初期値がモデルの解に近い値である程、モデルが解ける可能性は高くなる」**。従って、モデルの解の値について何らかの推測ができるのなら、その推測に基づき、できるだけ解の値に近い初期値を与えておくのが望ましい。上のプログラムで与えている初期値はある程度、解についての推測に基づき設定している[2)]。モデルが

2) 実際には前もって解がわかっているので、その解にある程度近い値を設定している。事前に解がわかっているというのは、そもそもこちらが指定した値が解になるようにモデルのパラメータや外生変数を決めているからである。

解けないときには、変数の初期値を変更してみるのが一つの対処方法である。

```
*       ----------------------------------------------------------------
*       ニュメレールの指定
p.fx("agr") = 1;
```

第 2 章で指摘したように、このモデルは価格について 0 次同次の性質を持つ。よって、絶対価格は不決定で、相対価格しか決まらない。通常このような場合にはある財をニュメレール（価値尺度財）に選択し、その価格を 1 に固定してモデルを解く。ここでは農産物（AGR）をニュメレールにしている。

変数に `fx` という接尾辞を付けると、変数をある値に固定する効果を持っている。MCP のモデルにおいて、このように変数の値を固定すると、その変数が対応している式がモデルから除外（ドロップ）されることになる。このモデルでは `p("agr")` という変数は `e_p("agr")` という式、つまり AGR の市場均衡条件に対応付られている。従って、`p("agr")` を 1 と固定することで、AGR の市場均衡条件がモデルから除外されることになる。市場均衡条件が 1 本少なくなるが、元々ワルラス法則より市場均衡条件の 1 本は redundant であるので、これでちょうど**「変数の数＝式の数」**となってモデルが解けることになる。このニュメレールの設定、及びワルラス法則については後でもう一度チェックをおこなう。

[注]

実際には、ニュメレールを指定しなくても GAMS ではモデルを解くことはできる。試しに上の 1 行をコメントアウトしてプログラムを実行してみて欲しい。しかし、モデルが解けるといっても、絶対価格が不定であることは変わらず、解として得られる絶対価格は無数にある解のうちの 1 つのパターンに過ぎないので、注意して欲しい。

```
*       ----------------------------------------------------------------
*       モデルを解く
$ontext
MCP としてモデルを解く。
$offtext
option mcp = path;
solve ge_sample_dual using mcp;
```

ここではモデルを解いている。「`option mcp = path;`」というのはソルバー（solver）の指定である。ここでは MCP を解くソルバーとして PATH を指定している。GAMS では一つのタイプのモデルに対して複数のソルバーが存在しているのが普通である。例えば、MCP モデルには PATH と MILES という 2 つのソルバーがあり、NLP モデルには CONOPT、MINOS、PATHNLP、BARON 等、多数のソルバーがある。MCP ソルバーとしては MILES よりも PATH の方が高性能

であるので、ここでは PATH を指定している[3)]。

3.3 計算結果

```
               S O L V E      S U M M A R Y

     MODEL   ge_sample_dual
     TYPE    MCP
     SOLVER  PATH                FROM LINE  298

**** SOLVER STATUS     1 Normal Completion
**** MODEL STATUS      1 Optimal

 RESOURCE USAGE, LIMIT          0.016 10000000000.000
 ITERATION COUNT, LIMIT      1205    2147483647
 EVALUATION ERRORS              0             0
51 row/cols, 183 non-zeros, 7.04% dense.
```

プログラムに間違いがなければ、正常に解けて、上のような結果が listing ファイルに出力されるはずである。SOLVER STATUS が 1 の Normal Completion、MODEL STATUS が 1 の Optimal となっていれば正常に解けている。もし他の数字、メッセージが出ていたらエラーが存在するか、もしくは正常に解けてはいないことを意味する。

```
---- VAR y  生産量

          LOWER          LEVEL          UPPER         MARGINAL

agr         .          140.0000        +INF             .
man         .          300.0000        +INF             .
ser         .          150.0000        +INF             .
```

これは生産量を表す変数 y の解である。このような値になっていたらプログラムに誤りはなく、正常に解けているはずである。

```
---- VAR p_f  生産要素の価格

          LOWER          LEVEL          UPPER         MARGINAL

lab   1.0000000E-6       1.0000        +INF       2.0514335E-8
cap   1.0000000E-6       1.0000        +INF       1.0476839E-8
```

3) もしこのように明示的に設定をしないのなら、デフォルトの設定が利用される。

```
                              LOWER          LEVEL          UPPER         MARGINAL

---- VAR u                      .          360.0000        +INF             .
---- VAR m                      .          360.0000        +INF             .

  u  効用水準
  m  所得
```

これは生産要素価格 p_f、効用水準 u、所得水準 m の解である。p_f の値が労働、資本の両方について 1 になっているが、これは偶然ではない。そもそも均衡価格が 1 になるようにパラメータや外生変数を設定したためである。p や p_va についてもやはり 1 になっているはずであるので、確認して欲しい。

表 4：計算結果

変数	記号		AGR	MAN	SER	LAB	CAP	その他
生産量	y(i)		140	300	150			
合成生産要素	v_a(i)		80	200	80			
単位投入需要	a_x(j,i)	AGR	0.21	0.03	0.20			
		MAN	0.07	0.17	0.13			
		SER	0.14	0.13	0.13			
投入需要	x(j,i)	AGR	30	10	30			
		MAN	10	50	20			
		SER	20	40	20			
単位合成生産要素需要	a_v(i)		0.57	0.67	0.53			
単位生産要素需要	a_f(f,i)	LAB	0.63	0.40	0.63			
		CAP	0.38	0.60	0.38			
生産要素需要	vf(f,i)	LAB	50	80	50			
		CAP	30	120	30			
消費需要	d(i)		70	220	70			
財の価格	p(i)		1	1	1			
合成生産要素の価格	p_va(i)		1	1	1			
生産要素の価格	p_f(f)					1	1	
効用水準	u							360
所得	m							360

表 4 が計算結果における全ての変数の値である。自分で解いたモデルの解と一致しているか確

認してみて欲しい。

とりあえず最初の設定の下で一度モデルを解いた後に、例として一つだけシミュレーションをおこなっている。具体的には、資本の賦存量を 20% 減少させるというシミュレーションである。以下が、その部分のプログラムである。

```
*       --------------------------------------------------------------
*       資本の賦存量が 20% 減少するシミュレーション
v_bar("cap") = v_bar0("cap") * 0.8;

solve ge_sample_dual using mcp;

*       元の値に戻しておく。
v_bar("cap") = v_bar0("cap");
```

資本の賦存量は v_bar("cap") というパラメータの値で表されていた。従って、それを 20%減少させ、モデルを解いている。モデルを解いた後に再び初期値に戻している。モデルを解いた結果は次の通りである。

```
---- VAR y  生産量

           LOWER          LEVEL          UPPER         MARGINAL

agr          .           127.3270        +INF            .
man          .           263.0791        +INF            .
ser          .           136.0850        +INF            .
```

3 つの財とも生産量は初期均衡の値から減少している。生産要素の賦存量を減少させたのでこの結果は当然であろう[4]。財の中では特に MAN の生産の減少率が最も大きい。これは MAN が最も資本集約的な産業であるからであろう。

```
---- VAR p_f  生産要素の価格

           LOWER          LEVEL          UPPER         MARGINAL

lab   1.0000000E-6         0.8272        +INF            .
cap   1.0000000E-6         1.2924        +INF            .

                            LOWER          LEVEL          UPPER         MARGINAL
```

4) ただし、ある一つ財の生産は増加するというケースもあるかもしれない。

```
---- VAR u                  .            320.0000        +INF            .
---- VAR m                  .            334.9961        +INF            .

  u  効用水準
  m  所得
```

ここは p_f、u、m の値である。当然だが初期均衡値からはどれも変化している。労働の価格（賃金）は低下しているのに対し、資本の価格は上昇している。これは資本の供給が減少したためであろう。ただし、この価格の水準は AGR をニュメレールとしたという前提での価格であることに注意して欲しい。他の財をニュメレールにとっていたら違う値になる。効用水準は低下している。生産要素の賦存量を減少させたのでこれは当然の結果である。

4　終わりに

本章では一般均衡モデルを GAMS で解くという作業をおこなった。具体的には第 2 章で紹介した一般均衡モデルを第 4 章で説明した方法で解いてみた。サンプルとして利用した ge_sample_dual.gms というプログラムは簡単に実行できるであろうが、これは元々問題なく実行できるように作成されたプログラムだからである。実際に、自分で一般均衡モデル（あるいは、CGE モデル）を作成し、プログラムを書くとなると、それほど簡単にはいかない。次の第 6 章では、一般均衡モデルを GAMS で作成する際にモデルをチェックする方法について説明する。

参考文献

Simon, C. P. and L. Blume (1994). *Mathematics for Economists*. New York: W. W. Norton Company.

5　履歴

- 2023-04-28: 誤植の修正。
- 2022-01-13: 説明の追加・修正。
- 2018-07-20: 説明の追加・修正。
- 2017-03-15: 説明の修正。