

## 応用一般均衡分析入門

第 6 章：一般均衡モデルのチェック  
方法 \*

武田 史郎 †

Date: 2023/05/08,

Version 4.0

## 目次

1	導入	2
2	マクロの説明	2
3	均衡価格の 0 次同次性（ニューメール）のチェック	3
3.1	プログラムの説明	3
3.2	結果	10
3.3	まとめ（価格についての 0 次同次性のチェック）	13
4	ワルラス法則のチェック	13
4.1	プログラムとシナリオ	13
4.2	結果	15
4.3	ワルラス法則が満たされていないケース	15
4.4	まとめ（ワルラス法則のチェック）	17
5	外生変数の比例変化によるチェック	17
5.1	シナリオ	17
5.2	結果	18
5.3	まとめ	18
6	簡単な政策シミュレーション	19
6.1	シナリオ	19
6.2	結果	20
6.2.1	シミュレーション結果	20
7	終わりに	21
	参考文献	22
8	問題	22

\*このファイルの配布場所: <https://shirotakena.github.io/ja/research-ja/cge-howto.html>†所属: 京都産業大学経済学部. Website: <https://shirotakena.github.io/ja/>

## 1 導入

今回の内容について。

- 第 5 章で一般均衡モデルを GAMS で表現して解いた。この第 6 章では第 5 章のモデルをチェックするという作業、及び簡単な政策のシミュレーションをおこなう。
- CGE 分析で利用するモデルは規模が大きく、変数も式も多くなる。そのようなモデルを表すプログラムは複雑になる。そのため自分の意図通りのモデルをスムーズにプログラムで表現できるということは少なく、プログラムを記述する段階で多くの間違いが入ってしまうことが多い。
- そこで自分の意図したモデルを作成できたかどうかをチェックする作業が必要になる。これには様々なアプローチがあるが、第 6 章では次の 3 つのチェックをおこなう。
  - 均衡価格の 0 次同次性（ニューメレール）に関するチェック
  - ワルラス法則に関するチェック
  - 外生変数の比例変化によるチェック
 3 つとも我々のモデルにおいて成り立っていないと判断しなければならない性質が実際に成り立っているかどうかを確認することで、モデル、データ、プログラムの正しさをチェックするという方法である。
- また、政策分析の簡単な例として、次の 2 つのシミュレーションをおこなう。
  - 一部の部門に対して 10 % の消費税を導入するシミュレーション
  - 3 つの部門の労働投入に 20 % の課税を導入するシミュレーション
- モデル自体の説明は第 5 章でおこなっているのので、そちらを参照して欲しい。

## 2 マクロの説明

この第 6 章で利用するプログラムでは、GAMS の「マクロ」という機能を利用している。モデルのチェックの前にまずマクロとはどのようなものかを簡単に説明しておく。マクロとは、ある一定の表現（手順）を何度も利用するための機能である。

マクロの基本的な書式は次の通りである。

```
$macro macro_name macro_body
```

マクロは \$macro 命令によって定義する。macro\_name の部分がマクロの名前になり、macro\_body のところに記述されたプログラムがマクロの中身になる。例えば、以下のプログラム

---

では `ratio` という名前のマクロを定義している。

```
parameter a, b, c;  
a = 1;  
b = 2;  
$macro ratio(x,y) x / y  
c = ratio(a,b);  
display c;
```

`x` と `y` はマクロの引数である。マクロはプログラムを実行する際には中身が展開される。よって、`c` に `ratio(a,b)` の値を代入する部分は次のように展開されて実行される。

```
c = a / b;
```

つまり、`ratio(a,b)` と書けば、`a / b` と書いたときと同じになるということである。プログラムを書く際、煩雑で複雑な表現を何度も利用するというのがよくある。そのような場合には、その表現をマクロに定義することでプログラムを簡潔にすることができる。非常に便利な機能であるので、利用法を覚えておくのがよい。マクロの使い方について詳しくは“[Macros in GAMS](#)”を見て欲しい。

### 3 均衡価格の 0 次同次性（ニューメレール）のチェック

第 2 章において説明したように、我々が考える一般均衡モデルでは、均衡価格の 0 次同次性という性質が成り立った。また、その性質を基に、ある財をニューメレールとしてモデルを解くという方法が用いられた。この均衡価格の 0 次同次性が実際に成り立っているかをチェックする。

#### 3.1 プログラムの説明

`ge_zero_h_check.gms` はニューメレールに関するチェックをおこなうプログラムである。以下、内容を説明していく。

```
*      ge_sample_dual.gms を読み込む。  
$include ge_sample_dual.gms
```

`ge_zero_h_check.gms` ではモデルには第 5 章の `ge_sample_dual.gms` のモデルをそのまま利用する。よって、まずここで `ge_sample_dual.gms` をそのまま読み込んでいる。`$include` はファイルを読み込むという命令である。実行の際にここで「282 Unable to open include file」というよう

なエラーが生じたら、読み込むファイルが見つからないということである<sup>1)</sup>。

```
parameter
  results      結果を表示するためのパラメータ
  results_pc   変化率を表示するためのパラメータ (%)
;
```

ここではまず `results` と `results_pc` というパラメータを宣言している。これは計算結果を入れておくためのパラメータである。

```
$macro out_results(sc) \
  results("u",sc) = u.l; \
  results("y_agr",sc) = y.l("agr"); \
  results("y_man",sc) = y.l("man"); \
  results("y_ser",sc) = y.l("ser"); \
  results("c_agr",sc) = d.l("agr"); \
  results("c_man",sc) = d.l("man"); \
  results("c_ser",sc) = d.l("ser"); \
  results("p_man",sc) = p.l("man"); \
  results("pf_lab",sc) = p_f.l("lab"); \
  results("m",sc) = m.l; \
  results("p_man-p_agr",sc) = p.l("man") / p.l("agr"); \
  results("pf_lab-p_agr",sc) = p_f.l("lab") / p.l("agr"); \
  results("m-p_agr",sc) = m.l / p.l("agr"); \
  results("p_man-pf_lab",sc) = p.l("man") / p_f.l("lab"); \
  results("pf_lab-pf_lab",sc) = p_f.l("lab") / p_f.l("lab"); \
  results("m-pf_lab",sc) = m.l / p_f.l("lab");
```

ここで `out_results` というマクロを定義している<sup>2)</sup>。これは計算結果の様々な値を `results` というパラメータに代入する動作を一つのマクロとしてまとめたものである。この `out_results` というマクロは `sc` という引数を持っているが、`sc` には後にシナリオを表す集合を入れる。

行末の「バックスラッシュ (\)」は記述が次の行に継続しているということを表す記号である。上のプログラムは合計 10 数行あるが、それをまとめて一つの `out_results` というマクロとして定義するためにバックスラッシュを利用している。このように一度 `out_results` を定義すれば、後は `out_results` とプログラムに書くことで、上の 10 数行のプログラムを実行するのと同じ動作をすることになる。

以下、マクロを使って `results` に代入している値を説明しておく。

```
results("u",sc) = u.l; \
```

- 1) その場合、GAMS IDE を利用しているのなら、タイトルバーに表示されているプロジェクトフォルダにファイルがあるかどうかを確認する。他の場合は、2つの GMS ファイルが同じフォルダにあるかを確認する。
- 2) バックスラッシュは場合によっては ¥マークに表示される。

これは効用の水準を表している。変数のレベル値を代入するので、どの変数にも .1 を付けている。

```
results("y_agr",sc) = y.l("agr"); \
results("y_man",sc) = y.l("man"); \
results("y_ser",sc) = y.l("ser"); \
```

一番上は AGR 部門の生産量である。他の 2 行は MAN、SER の生産量である。

```
results("c_agr",sc) = d.l("agr"); \
results("c_man",sc) = d.l("man"); \
results("c_ser",sc) = d.l("ser"); \
```

一行目は AGR 財の消費量である。このモデルでは中間投入もあるので、生産量  $\neq$  消費量である。他の 2 行は MAN、SER の消費量である。

```
results("p_man",sc) = p.l("man"); \
```

これは MAN の価格である。p("man") のそのままの値であるので、以下ではこの価格を便宜上「絶対価格」と呼ぶことにする。ただし、この価格は AGR をニューメレールとし、その価格を 1 と設定したときの p("man") の価格であるので、厳密には AGR の価格を 1 に固定した上での  $p(\text{"man"})/p(\text{"agr"})$  という相対価格でもある。

```
results("pf_lab",sc) = p.f.l("lab"); \
results("m",sc) = m.l; \
```

これは労働の価格と所得水準である。これも以下では絶対価格・所得と呼ぶ。

```
results("p_man-p_agr",sc) = p.l("man") / p.l("agr"); \
```

これは AGR で測った MAN の相対価格であり、MAN 一つが AGR 何個分の価値を持つかを表している。元々、AGR をニューメレールにとりその価格を 1 としているので、この相対価格は単なる  $p.l(\text{"man"})$  と同じ値をとる。しかし、ニューメレールを変更した場合には、 $p.l(\text{"man"})$  と  $p.l(\text{"man"})/p.l(\text{"agr"})$  の値は変わってくる。実際、以下ではニューメレールを変更するシミュ

レーションをおこない、どう変わるかをチェックする。

```
results("pf_lab-p_agr",sc) = p_f.l("lab") / p.l("agr"); \
results("m-p_agr",sc) = m.l / p.l("agr"); \
results("p_man-pf_lab",sc) = p.l("man") / p_f.l("lab"); \
results("pf_lab-pf_lab",sc) = p_f.l("lab") / p_f.l("lab"); \
results("m-pf_lab",sc) = m.l / p_f.l("lab");
```

以上は全て相対価格である。p("agr") で割っているのは AGR で測った相対価格、p\_f("lab") で割っているのは労働で測った相対価格である。最初の二つは AGR で測った相対価格、下の二つは労働で測った相対価格である。最後は労働で測った相対所得（実質所得）である。

それではプログラムの続きを説明していこう。

```
set var 結果表示用の集合 /
    u, y_agr, y_man, y_ser, c_agr, c_man, c_ser, p_man, pf_lab, m,
    p_man-p_agr, pf_lab-p_agr, m-p_agr, p_man-pf_lab,
    pf_lab-pf_lab, m-pf_lab /;
```

ここでは計算結果をまとめるために利用する集合 var を定義している。

```
set      sce      計算するシナリオ /
          sce1     デフォルトの均衡
          sce2     資本の賦存量を2倍にした均衡
          sce3     AGRの価格を2倍にした均衡
          sce4     ニュメレールを労働に変更したときの均衡
          sce5     SCE2+ニュメレールを労働に変更したときの均衡
          /;
display sce;
```

ここではシナリオを表すための集合 sce を定義している。sce の要素は sce1~sce5 の5つであり、以下ではシナリオとして5つを考える。

#### [シナリオ sce1]

これはデフォルトの均衡である。ge\_sample\_dual.gms のプログラムと同じ均衡である。これを比較の基準とする。

#### [シナリオ sce2]

これは資本の賦存量を2倍にしたときの均衡である。このシナリオでは各財の生産量、価格がど

う変わるかをチェックする。

#### [シナリオ sce3]

これは AGR の価格を 2 倍にした均衡である。元々、AGR をニューメレールとして、その価格を 1 に固定していた。ここではその価格を 2 に変更してみる。価格について 0 次同次性が成り立つモデルであるので、AGR の価格を (1 から) 2 に変更すれば、数量変数や相対価格は全く変わらず、単に絶対価格が全て 2 倍になるだけのはずである。このシナリオではこれをチェックする。

#### [シナリオ sce4]

ニューメレールを労働に変更する。つまり、労働の価格を 1 に固定して計算する。ニューメレールの変更によって、均衡がどう変化するかをチェックする。価格について 0 次同次性が満たされるモデルではニューメレールの選択は数量変数の値、相対価格の値には全く影響を与えないはずである。それを確認する。

#### [シナリオ sce5]

sce4 の設定で、資本の賦存量を 2 倍にするというシミュレーションをおこなう。ニューメレールを変更することで、sce2 の場合と結果がどう変わるかをチェックする。0 次同次性が満たされるなら、sce2 のケースとは絶対価格の値は変わってくるが、相対価格、数量変数の値は同じになるはずである。

以下は各シナリオを解くプログラムである。

```
* -----
*   SCE1: デフォルトの設定でのシミュレーション
*
*   モデルを解く。
solve ge_sample_dual using mcp;
*
*   以下はマクロの実行。
out_results("sce1");
```

ここではデフォルトのシナリオを解いている。特に何の変更もせず、solve 命令によりモデルを解いているだけである。out\_results("sce1") という命令によって、マクロ our\_results の引数 sc の部分に "sce1" が入る形で、マクロが実行されることになる。

```
* -----
*   SCE2: 資本の賦存量を 2 倍にした均衡
*   2 倍に増加
v_bar("cap") = v_bar0("cap") * 2;
*
solve ge_sample_dual using mcp;
```

```

*      マクロの実行
out_resutls("sce2");

*      元の値に戻す。
v_bar("cap") = v_bar0("cap");

```

ここは sce2 を解く部分である。資本の賦存量を 2 倍にして、モデルを解いている。解いた後は先ほどと同様マクロを実行し、さらにここでは資本の賦存量を元々の値に戻している。

```

*      -----
*      SCE3: AGR の価格を 2 倍にした均衡
p.fx("agr") = 2;

solve ge_sample_dual using mcp;

out_resutls("sce3");

p.fx("agr") = 1;

```

AGR の価格は元々 1 に固定されているが、この部分ではそれを 2 に変更してモデルを解いている。後は同様である。

```

*      -----
*      SCE4: ニュメレールを労働に変更したときの均衡
p.up("agr") = inf;
p.lo("agr") = 0;
p_f.fx("lab") = 1;

solve ge_sample_dual using mcp;

out_resutls("sce4");

```

ニュメレールを AGR から労働に変更している。具体的には、これまで 1 に固定していた `p("agr")` を自由に動くように変更し（ただし、下限は 0 としている）、代わりに `p_f("lab")` を 1 に固定している。

```

*      -----
*      SCE5: sce4+資本の賦存量を 2 倍にした均衡
p.up("agr") = inf;
p.lo("agr") = 0;

```

```

p_f.fx("lab") = 1;

v_bar("cap") = v_bar0("cap") * 2;

solve ge_sample_dual using mcp;

out_results("sce5");

v_bar("cap") = v_bar0("cap");

```

ここでは、ニュメールを労働にした上で、sce2 と同様のシミュレーションをおこなっている。

```

* -----
*      結果のまとめ
*
*      results パラメータの中身を表示
display results;

*      デフォルトのシナリオの値からの変化率を求める。
results_pc(var,sce)$results(var,"sce1")
    = 100 * (results(var,sce)/results(var,"sce1") - 1);

*      1e-6 で数値をまるめる。
results_pc(var,sce) = round(results_pc(var,sce), 6) + eps;

*      results_pc パラメータの中身を表示
display results_pc;

```

ここは計算結果をまとめている部分である。results に変数の値を代入したが、それから変数の変化率 (sce1 の値から何%変化したか) を求め、results\_pc に代入している。

```

execute_unload "ge_zero_h_check.gdx", results, results_pc;

```

ここはパラメータ results と results\_pc の値を ge\_zero\_h\_check.gdx という GDX ファイルにエクスポートしている。execute\_unload という命令は以下のような書式を持つ。

```

execute_unload "GDX ファイル名", 出力するもの 1, 出力するもの 2;

```

“GDX ファイル名”の部分に出力する GDX ファイルの名前を指定する。そして、その後に、出力するものを列挙する (複数あるときはカンマで区切る)。出力可能なのは、パラメータ、変数、

集合などである。

`execute_unload` 命令については詳しくは“[GAMS Data eXchange \(GDX\)](#)”を参照して欲しい。

```
*      $onecho ~ $offecho の間の部分を temp.txt というファイルの中へ出力
$onecho > temp.txt
i=ge_zero_h_check.gdx o=ge_zero_h_check.xlsx epsout=0
par=results rng=results!A1 rdim=1 cdim=1
par=results_pc rng=results_pc!A1 rdim=1 cdim=1
$offecho

*      temp.txt の中身を引数として gdxxrw を実行する。
execute '=gdxxrw @temp.txt';
*      temp.txt を削除
execute 'del temp.txt';
```

ここでは GDX ファイル内の `results` と `results_pc` というパラメータの値を Excel ファイル `ge_zero_h_check.xlsx` にエクスポートしている。まず、GDX ファイルから Excel ファイルに出力する設定を `temp.txt` というファイルに書き込み、その `temp.txt` というファイルの中身を引数として `gdxxrw` を実行するという手順になっている。

- 「`$onecho > temp.txt`」という命令で `$onecho`～`$offecho` の間の記述を `temp.txt` というファイルに出力することになる。
- 「`gdxxrw @temp.txt`」というように `gdxxrw` の後に `@` マークを付けたファイル名を書くと、`temp.txt` の中身を引数として `gdxxrw` を実行せよという意味になる。
- `execute` は外部プログラムを実行する命令であるが、「`execute '=gdxxrw'`」のように外部プログラム名の前に「`=`」を付けると、そのプログラムの実行が終了するまでその後のコードを実行せずに待てという意味になる。
- `execute 'del temp.txt';` は `temp.txt` ファイルを削除するという意味である。「`del`」は Windows に標準的に付属する、ファイルを削除するためのプログラム名である。`execute` 命令、外部プログラム `gdxxrw` については詳しくは“[GDXXRW](#)”を参照して欲しい。

## 3.2 結果

それでは結果を見てみよう。表 1 と表 2 は上のプログラムを実行した結果を `ge_zero_h_check.xlsx` に出力したものである。表 1 は各シナリオにおける変数の値（水準）を表しており、表 2 は `sce1` の値からの変化率（%）を表している。

[`sce1`]

均衡において全ての絶対価格は 1 になる。これは `ge_sample_dual.gms` で既に見た。こうなる

表 1: 変数の値

		sce1	sce2	sce3	sce4	sce5
効用	u	360.00	480.00	360.00	360.00	480.00
	y_agr	140.00	174.80	140.00	140.00	174.80
生産量	y_man	300.00	417.06	300.00	300.00	417.06
	y_ser	150.00	188.56	150.00	150.00	188.56
消費量	c_agr	70.00	87.40	70.00	70.00	87.40
	c_man	220.00	305.85	220.00	220.00	305.85
	c_ser	70.00	88.00	70.00	70.00	88.00
絶対価格	p_man	1.00	0.81	2.00	1.00	0.52
	pf_lab	1.00	1.56	2.00	1.00	1.00
	m	360.00	420.89	720.00	360.00	270.00
AGRで測った価格	p_man-p_agr	1.00	0.81	1.00	1.00	0.81
	pf_lab-p_agr	1.00	1.56	1.00	1.00	1.56
	m-p_agr	360.00	420.89	360.00	360.00	420.89
労働で測った価格	p_man-pf_lab	1.00	0.52	1.00	1.00	0.52
	pf_lab-pf_lab	1.00	1.00	1.00	1.00	1.00
	m-pf_lab	360.00	270.00	360.00	360.00	270.00

表 2: sce1 における値からの変化率 (%)

		sce1	sce2	sce3	sce4	sce5
効用	u	0.00	33.33	0.00	0.00	33.33
	y_agr	0.00	24.85	0.00	0.00	24.85
生産量	y_man	0.00	39.02	0.00	0.00	39.02
	y_ser	0.00	25.71	0.00	0.00	25.71
消費量	c_agr	0.00	24.85	0.00	0.00	24.85
	c_man	0.00	39.02	0.00	0.00	39.02
	c_ser	0.00	25.71	0.00	0.00	25.71
絶対価格	p_man	0.00	-19.34	100.00	0.00	-48.26
	pf_lab	0.00	55.88	100.00	0.00	0.00
	m	0.00	16.91	100.00	0.00	-25.00
AGRで測った価格	p_man-p_agr	0.00	-19.34	0.00	0.00	-19.34
	pf_lab-p_agr	0.00	55.88	0.00	0.00	55.88
	m-p_agr	0.00	16.91	0.00	0.00	16.91
労働で測った価格	p_man-pf_lab	0.00	-48.26	0.00	0.00	-48.26
	pf_lab-pf_lab	0.00	0.00	0.00	0.00	0.00
	m-pf_lab	0.00	-25.00	0.00	0.00	-25.00

のは、元々、均衡価格が1になるようにモデルのパラメータを設定しているためである。

[sce2]

資本の賦存量を2倍にした結果、生産量は全部門で増加している。特にMANの生産の増加率が高い。これはMANが資本集約的な部門であるためである。MANの絶対価格は低下している。これはMANの生産の伸び率が大きいことが原因である。一方、労働の絶対価格はかなり上昇している。これは労働が相対的に希少になったことが原因と考えられる。

効用も上昇している。これは要素賦存量が増加しているのが当然の結果である。絶対的な所得、及びAGRで測った相対所得は上昇しているのに対し、労働で測った相対所得は低下している。労働で測った相対所得が低下しているのは労働の価格の上昇が高いためである。このように、何を基準にして測るかによって価格、所得の変化の大きさ、方向は変わってくる。これは、計算結果を見るとときに、何をニューメレルにとっているのか、あるいは何を基準にしているのか（何をデフレーターにしているのか）が重要であるということである。

[sce3]

生産量、消費量、効用など数量を表す変数についてはsce1のときと全く変わらない。また、AGRで測った相対価格、労働で測った相対価格も全く変わらない。変わるのは絶対価格だけであり、理論モデルから導けるように全て2倍になっている。このモデルでは実際に価格について0次同次性が成り立っていることが確認できる。

[sce4]

このケースはsce1と全く変わらない。つまり、ニューメレルを労働としても（労働の価格を1と固定して解いても）、AGRがニューメレルのときと全く変わらないということである。

数量変数や相対価格が何をニューメレルにしようと思わないのは当然の結果である。しかし、ここでは絶対価格、 $p(\text{"man"})$  や  $m$  などの値も変わっていない。本来、これらの値は何をニューメレルにするかによって変化するはずである。今回、 $p(\text{"man"})$  や  $m$  の値も変わっていないのは、労働の価格  $p_f(\text{"lab"})$  がsce1において元々1という値をとっていたためである。労働にニューメレルを変更しているが、それが元々均衡において1という値をとっていたものを1に固定したというだけであるので、全く何も変わらないという結果となっている。

[sce5]

sce4ではニューメレルを労働に変更したが、労働は元々1という均衡価格を持っていたので何も変わらないという結果となった。これではニューメレルを変更することの効果が変わりにくいのので、sce2と同じシミュレーションをニューメレルを変更した上でおこなったのがsce5である。ニューメレルの変更によってsce2とどのような違いが生じるかを確認する。

まず、効用、生産量、消費量などの数量変数の値は何も変わっていない。つまり、ニューメレルの変更は資本賦存量が数量変数に与える効果に全く影響を与えないということである。また、相対価格も全く変わっていない。ニューメレルの選択は相対価格にも全く影響しないことが確認できる。一方、絶対価格は変わっている。何をニューメレルにするかで絶対価格は全く違う値になるこ

とが確認できる。

### 3.3 まとめ（価格についての0次同次性のチェック）

以上、価格についての0次同次性が成り立っているかどうかを確認した。このプログラムでは実際に成り立っていることがわかったが、モデル、あるいはプログラムのどこかがおかしい場合には0次同次性が成り立たないということが生じてくる。例えば、`ge_zero_h_check_fail.gms` を実行して、結果を見て欲しい。このプログラムの実行結果では、ニュメレールの変更によって数量変数や相対価格の値が変わってしまう。これは均衡価格についての0次同次性が満たされないモデルになっているということであり、実際このプログラムには2カ所間違いが含まれている（自分で探してみたい）。

このように、ニュメレールを変更して解いてみて結果が変わらないかをチェックすることでモデル、プログラムに間違いがないかを確認することができる。簡単にできるチェックであるので、自分でモデルを作成したときには、必ずニュメレールを用いたチェックをおこなうのがよい<sup>3)</sup>。

## 4 ワルラス法則のチェック

価格についての0次同次性ととともに、我々が考える一般均衡モデルで成り立つ性質としてワルラス法則があった。以下では、ワルラス法則が実際に成り立っているかどうかをチェックする。

### 4.1 プログラムとシナリオ

モデルには第3節と同じモデルを利用する。`ge_walras_check.gms` がそのプログラムである。計算するシナリオは前節の `sce1`~`sce5` から `sce3` を除いた4つのシナリオである。以下、主に `ge_zero_h_check.gms` との相違点を説明する。

```
*      ワルラス法則が成立しているかのチェック。0になればOK
      results("p_agr_m",sc) = p.m("agr"); \
      results("pf_lab_m",sc) = p_f.m("lab"); \
```

ここはマクロ `out_results(sc)` の定義の部分である。ここでは `p("agr")` と `p_f("lab")` の marginal 値を代入している。第4章で説明したように、MCPタイプのモデルにおいて変数の marginal 値はその変数が対応する式の乖離幅を表している。モデルを定義している `ge_sample_dual.gms` を見て欲しい。例えば、変数 `p` は次のような式 `eq_p` に対応している。

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

3) 当たり前の話のように聞こえるかもしれないが、CGE分析の学習者には0次同次性が満たされないモデルを作成してシミュレーションをしようとし、正常に解けないという問題に直面していることがよく見られる。

この  $e_p(i)$  の定義より、変数  $p(i)$  の marginal 値は次の値を表すことになる。

$$y(i) - \sum(j, a_x(i,j)*y(j)) - d(i)$$

これは財  $i$  の超過供給量であるので、変数  $p(i)$  の marginal 値は結局財  $i$  の超過供給量を表すことになる。よって、 $p.m(i)$  がゼロかどうかを見ることで、財  $i$  の市場が均衡しているかどうか分かる。同様に変数  $p_f(f)$  は生産要素  $f$  の市場均衡条件  $e_{p_f}(f)$  に対応しており、 $p_f.m(f)$  の値は生産要素  $f$  の超過供給量を表すことになる。

```
*      これもワルラス法則が成立しているかのチェック（全ての市場の超過
*      需要額の和を計算）。0 になれば OK
      results("v_ed",sc) = \
          sum(i, p.l(i)*(sum(j, a_x.l(i,j)*y.l(j)) + d.l(i) - y.l(i))) \
          + sum(f, p_f.l(f)*(sum(i, a_f.l(f,i)*v_a.l(i)) - v_bar(f))) \
          + sum(i, p_va.l(i)*(a_v.l(i)*y.l(i) - v_a.l(i))); \
```

これは全ての市場の超過需要額の和の値を計算し、パラメータに代入している部分である。**ワルラス法則とは全ての市場の超過需要額の和が0になるという性質であった。**よって、この値が0になるならば、ワルラス法則が満たされていることになる。

```
*      四捨五入
      results("p_agr_m",sc) = round(results("p_agr_m",sc), 5) + eps; \
      results("pf_lab_m",sc) = round(results("pf_lab_m",sc), 5) + eps; \
      results("v_ed",sc) = round(results("v_ed",sc), 5) + eps;
```

ここは小数点以下第6位で四捨五入して桁数を減らしている。eps は 0 に近いが 0 ではない数値を表す特殊な記号である。eps を加えているのは後で Excel ファイルに結果を出力するためである。

それでは以下でシナリオの説明をおこなう。

#### [シナリオ sce1]

これは何も変化させない元々の均衡である。この均衡で全ての市場の超過需要額の和が 0 になっているかをチェックする。また、このシナリオでは AGR をニューメレールにとって、 $p(\text{"agr"})$  を 1 に固定している。このように変数を固定すると、その変数が対応している式はモデルから除外される。変数  $p(\text{"agr"})$  は  $e_{p}(\text{"agr"})$  に対応しているので、この場合は AGR の市場均衡条件がモデルから除外されることになる。しかし、もしワルラス法則が成立するならば、AGR 以外の

市場が均衡していれば、AGR の市場も自動的に均衡するはずである。ワルラス法則が成り立っているかどうかのチェックのもう一つの方法として、モデルから除外された AGR の市場が均衡しているかをチェックする。これには `p.m("agr")` の値が 0 になっているかどうかを調べればよい。

#### [シナリオ sce2]

これは資本の賦存量を 2 倍にするシナリオである。チェックする項目は `sce1` と同じである。

#### [シナリオ sce4]

このシナリオではニューメレールを労働に変更している。全ての市場の超過需要額の和が 0 になっているかどうかをチェックするのは `sce1` と同じ。ニューメレールは労働に変更したのでこの場合は労働市場の均衡条件がモデルから除外されている。そこで労働市場の均衡条件が実際に成り立っているかをチェックする。これには `p_f.m("lab")` の値が 0 になっているかを確認すればよい。

#### [シナリオ sce5]

「`sce4` + 資本の賦存量を 2 倍」というシナリオである。この場合のチェック項目は `sce4` と同じ。

## 4.2 結果

表 3 は各シナリオにおける変数の値を掲載している。モデルもシナリオも同じであるので、基本的には第 3 節のシミュレーションの結果と同じである。

#### [sce1]

まず `p("agr")` の marginal 値はゼロになっている。よって、AGR の市場は均衡していることが確認できる。これはワルラス法則が満たされていることを意味する。また、全ての市場の超過需要額の和はゼロになっている。こちらでもワルラス法則が満たされていることがわかる。

#### [sce2]

資本を増加させても `sce1` で成立した二つの結果が同様に成り立っていることがわかる。

#### [sce4 & sce5]

ニューメレールを変更したが、ニューメレールの市場は均衡しているし、超過需要額の和はゼロであることがわかる。やはりワルラス法則は成り立っていることがわかる。

## 4.3 ワルラス法則が満たされていないケース

前節のプログラムではワルラス法則は満たされた。ここでは参考のためワルラス法則が満たされないケースを見てみよう。以下は `ge_walras_check_fail.gms` を実行した結果である。このプログラムは基本的に `ge_walras_check.gms` と同じであるが、モデルのファイルとして `ge_sample_dual_alt.gms` を利用しているところだけが異なっている。

#### [結果]

結果は表 4 にある。この場合、`sce1` では `p("agr")` の marginal 値が  $-29.91$  になっている。こ

表3：ワルラス法則のチェック

		sce1	sce2	sce4	sce5
効用	u	360.00	480.00	360.00	480.00
生産量	y_agr	140.00	174.80	140.00	174.80
	y_man	300.00	417.06	300.00	417.06
	y_ser	150.00	188.56	150.00	188.56
消費量	c_agr	70.00	87.40	70.00	87.40
	c_man	220.00	305.85	220.00	305.85
	c_ser	70.00	88.00	70.00	88.00
絶対価格	p_man	1.00	0.81	1.00	0.52
	pf_lab	1.00	1.56	1.00	1.00
	m	360.00	420.89	360.00	270.00
Marginal値	p_agr_m	0.00	0.00	0.00	0.00
	pf_lab_m	0.00	0.00	0.00	0.00
超過需要額	v_ed	0.00	0.00	0.00	0.00

表4：ワルラス法則のチェック（上手くいかないケース）

		sce1	sce2	sce4	sce5
効用	u	359.88	486.61	356.16	508.13
生産量	y_agr	117.87	148.59	158.33	204.47
	y_man	311.13	435.07	310.36	455.25
	y_ser	159.88	198.73	167.37	218.16
消費量	c_agr	70.13	88.16	70.28	93.42
	c_man	219.64	311.10	215.75	321.69
	c_ser	70.11	88.81	70.17	93.97
絶対価格	p_man	1.01	0.79	1.18	0.56
	pf_lab	0.98	1.54	1.00	1.00
	m	361.44	422.43	413.27	307.23
Marginal値	p_agr_m	-29.91	-33.78	0.00	0.00
	pf_lab_m	0.00	0.00	-27.80	-35.71
超過需要額	v_ed	-0.16	6.00	-6.57	16.63

これは AGR の市場に 29.91 の超過需要が生じていることを表している。その結果、全ての市場の超過需要額の和もゼロにはなっていない。また、sce4 では労働市場に 27.80 の超過需要が発生している。sce2、sce5 についてもニューメレールの市場が均衡していないのと同時に、超過需要の総額が 0 になっていない。

以上の結果は、このモデル (ge\_sample\_dual\_alt.gms のモデル) においてワルラス法則が満たされていないことを示している。これはモデル、あるいはデータのどこかに間違いが含まれていることを意味している。

#### 4.4 まとめ (ワルラス法則のチェック)

ワルラス法則のチェックも、価格の 0 次同次性のチェック同様に、モデル、データ、プログラムの誤りをチェックするために利用できる。価格の 0 次同次性のチェック同様に、自分でモデルを作成したときには、このチェックをおこなうのがよい。

## 5 外生変数の比例変化によるチェック

ここまで我々が考えてきたモデル (この後のモデルでも同じだが) は生産関数や効用関数に一次同次の関数を利用している。このようなモデルにおいては、モデルの外生変数 (主に生産要素の賦存量) を同率、例えば X% だけ比例的に変化させたときに

- 全ての内生的な数量変数 (生産量、消費量等)、金額変数 (生産額、消費額、所得額、支出額等) も同率で (X%) 変化する
- 価格変数のみ変わらない

という結果になるはずである。逆に言えば、このような結果が成り立たないならばモデル、データ、プログラムのどこかがおかしいということになる。「外生変数の比例変化によるチェック」とはこの性質を利用しておこなうチェックのことである。

[注意]: 効用関数や生産関数に一次同次関数を利用していないなら元々上の性質は成り立たないので、このチェックは使えない。費用に固定費用がかかるモデルや、従量税を導入しているモデル、数量規制を導入しているモデルでも同じである。

### 5.1 シナリオ

ge\_proportional\_check.gms というファイルがこのチェックをおこなうプログラムである。中身は価格の 0 次同次性をチェックした ge\_zero\_h\_check.gms とほぼ同じである。ただし、解くシナリオだけ違っている。シナリオは次の 4 つである。

- sce1: これは何も変化させないで解くシナリオ。
- sce2: これは資本の賦存量を 2 倍にするシナリオ
- sce3: これは資本と労働の賦存量を 2 倍にするシナリオ
- sce4: これは資本と労働の賦存量を 0.5 倍にするシナリオ

チェックしたいのは sce3 において数量変数が 2 倍になると同時に、価格が変化しないかどうか、sce4 において数量変数が 0.5 倍になると同時に、価格が変化していないかどうかということである。

## 5.2 結果

表 5：外生変数の比例変化によるチェック（sce1 からの変化率、%）

		sce1	sce2	sce3	sce4
効用	u	0.00	33.33	100.00	-50.00
	y_agr	0.00	24.85	100.00	-50.00
生産量	y_man	0.00	39.02	100.00	-50.00
	y_ser	0.00	25.71	100.00	-50.00
消費量	c_agr	0.00	24.85	100.00	-50.00
	c_man	0.00	39.02	100.00	-50.00
	c_ser	0.00	25.71	100.00	-50.00
絶対価格	p_man	0.00	-19.34	0.00	0.00
	pf_lab	0.00	55.88	0.00	0.00
	m	0.00	16.91	100.00	-50.00
	p_man-p_agr	0.00	-19.34	0.00	0.00
AGRで測った価格	pf_lab-p_agr	0.00	55.88	0.00	0.00
	m-p_agr	0.00	16.91	100.00	-50.00
労働で測った価格	p_man-pf_lab	0.00	-48.26	0.00	0.00
	pf_lab-pf_lab	0.00	0.00	0.00	0.00
	m-pf_lab	0.00	-25.00	100.00	-50.00

表 5 の値は sce1 からの変数の変化率 (%) を表している。まず、sce2 では生産量、消費量等の数量変数も価格もどちらも変化しており、その変化率は変数によって異なっている。これに対し、資本・労働をどちらも 2 倍ずつに増加させたシナリオでは、全ての数量変数、金額変数が 100% 増加（つまり、2 倍に増加）し、一方、価格変数は変化していない。期待通りの計算結果になっていることがわかる。sce4 についても同様に資本・労働の減少率に等しいだけ全ての数量変数、金額変数が減少し、価格変数は変化していない。

### 同率の変化をもたらさないケース

ge\_proportional\_check.gms では外生変数の比例的変化に対し、内生的な数量変数、金額変数は同率で変化し、価格変数は不変であった。ge\_proportional\_check\_fail.gms はそのような結果が成り立たないケースである。このプログラムではモデルのファイルとして ge\_sample\_dual\_alt.gms を用いている。実際、計算して見てチェックして欲しい。

## 5.3 まとめ

以上の例が示すように、ある種のモデルにおいては、外生変数を比例的に変化させるというシミュレーションをおこなうことでモデルの間違いをチェックすることが可能である。特に、この方法を使うことで、ニュメールの変更によるチェック、ワルラス法則によるチェックでは見つけれ

れない誤りを見つけることもできるので、ニューメーラールの変更によるチェック、ワルラス法則によるチェックとともにこの方法によるチェックをおこなうようにするとよい。

## 6 簡単な政策シミュレーション

この節では、例として次の二つの政策の分析をおこなう。

- MAN と SER に対して 10% の消費税を導入するシミュレーション
- 3 つの部門の労働投入に 20% の課税を導入するシミュレーション

### 6.1 シナリオ

ge\_simulation.gms というプログラムがこの政策のシミュレーションをおこなっているファイルである。利用するモデルのファイルは、ここまでと同様に ge\_sample\_dual.gms である。以下では、各シナリオの設定をしている部分のプログラムを説明する。

```
set      sce      計算するシナリオ /
              sce1   デフォルトの均衡
              sce2   man と ser に対して 10%の消費税を導入するシナリオ
              sce3   労働投入に 20%の課税を導入するシナリオ
              /;
display sce;
```

ここではシナリオを表す集合を定義している。シミュレーションではデフォルトの均衡から政策によってどれだけ変数が変化するかを分析する。

```
*      -----
*      SCE2: man と ser に対して 10%の消費税を導入するシナリオ
t_c("man") = 0.1;
t_c("ser") = 0.1;

solve ge_sample_dual using mcp;

out_resutls("sce2");

t_c(i) = 0;
```

ここでは sce2 を解いている。t\_c(i) が財 i の消費に対する従価税率を表すパラメータである。MAN と SER についてこの t\_c(i) に 0.1 をセットすると、両財の消費に 10% の従価税が課されることになる。元々 t\_c(i) はゼロに設定されている。つまり元々は消費税はないと仮定されてい

る。解いた後に初期値のゼロを入れ直している。

```

* -----
*      SCE3: 労働投入に 20%の課税を導入するシナリオ
t_f("lab",i) = 0.2;

solve ge_sample_dual using mcp;

out_resutls("sce3");

t_f(f,i) = 0;

```

ここでは sce3 を解いている。t\_f(f,i) は部門  $i$  の生産要素  $f$  の投入に対する税率を表している。t\_f("lab",i) に 0.2 を設定すると、全ての部門の労働投入に 20% の課税を導入することを意味する。ここでも解いた後に、初期値である 0 に戻している。

## 6.2 結果

シミュレーション結果を見る前に、各シナリオでのチェック項目を確認する。

[sce2]

これは MAN と SER に消費税を導入するシナリオである。MAN と SER は税金がかかるので、その消費、生産が影響を受けると考えられる。消費、生産がどう変化するかをチェックする。同時に、税金がかからない AGR の消費、生産がどう変化するかもチェックする。また、税金の導入によって効用の水準がどう変化するかもチェックする。

[sce3]

これは労働投入への税金という政策であるので、労働投入がどう変化するかをチェックする。また、その結果、効用水準がどう変化するかをチェックする。

それでは以下で結果を見てみよう。表 6 が結果を示している。

### 6.2.1 シミュレーション結果

表 6 の数値は sce1 における変数の数値からの変化率 (%) を表している。

[sce2]

消費税が導入された MAN と SER の消費量は減少し、それに伴い生産量を減少している。一方、消費税がかからない AGR の消費量、生産量は増加している。これは MAN と SER で利用されていた生産要素が AGR の生産に利用されるようになるからである。AGR で測った MAN の相対価格、労働で測った MAN の相対価格はともに低下している。これは MAN への消費税によって MAN への需要が減少したためである。効用水準は低下している。このモデルは完全競争モデル

表 6：政策シミュレーション (sce1 からの変化率、%)

		sce1	sce2	sce3
効用	u	0.00	-0.04	0.00
	y_agr	0.00	2.26	0.00
生産量	y_man	0.00	-0.74	0.00
	y_ser	0.00	-0.42	0.00
消費量	c_agr	0.00	3.85	0.00
	c_man	0.00	-0.92	0.00
	c_ser	0.00	-0.97	0.00
絶対価格	p_man	0.00	-0.12	0.00
	pf_lab	0.00	0.31	-16.67
	m	0.00	7.90	0.00
AGRで測った価格	p_man-p_agr	0.00	-0.12	0.00
	pf_lab-p_agr	0.00	0.31	-16.67
	m-p_agr	0.00	7.90	0.00
労働で測った価格	p_man-pf_lab	0.00	-0.43	20.00
	pf_lab-pf_lab	0.00	0.00	0.00
	m-pf_lab	0.00	7.57	20.00

で、他に何の歪みもないモデルである。よって、自由放任がパレート効率的であり、税金の導入は代表的家計の効用を低下させる方向に働くはずである。計算結果は実際にそのような結果が成り立っていることを示している。

#### [sce3]

表の結果からわかるように価格が変化するだけで生産量も消費量も変わらない。生産量が変わらないので労働投入量も変わらない。数量が全く変化しないので効用水準も全く変わらない。

なぜ税金を導入したにもかかわらずその効果がないのか？このモデルでは労働の供給量は外生的に固定されている。従って、どう税金をかけようが労働供給量全体が変化することはない。一方、労働課税は全ての部門で同じ率で導入されているので、部門間での労働の配分が変わる効果も生じない。以上のように、労働供給全体の量が変わらないことに加え、部門間での配分も変化しないので、全く何の効果も持たないという結果になる。

投入側で労働への課税を支払い、それを家計が受け取るという効果が生じているのであるが、労働課税の分、企業が支払う賃金、家計が受け取る賃金が減少する。この賃金の減少が家計に税の収入が生じるという効果を完全に相殺してしまうので、所得の変化の効果も生じない。

## 7 終わりに

この章では一般均衡モデルを数値的に解く際の、モデルのチェック方法を説明した。一般均衡モデルにおいて、成り立つべき性質が実際に成り立っているかを見ることで、モデルに誤りが含まれているかどうか判断するという方法であった。後に、第 9 章ではこの章で紹介した方法も含め、

CGE モデルをチェックする方法について見る。

## 参考文献

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

## 8 問題

### 問題 1

ここまでチェックのためのシミュレーションをおこなうにあたって、モデルとしては `ge_sample_dual.gms` のモデル、つまり双対アプローチを用いて記述したモデルを用いてきた。同じシミュレーションを伝統的アプローチによって記述したモデルを用いておこなってみよ。

### 問題 2

全ての財に同率（例えば、10%）の消費税を課すという政策の効果を計算し、その結果を説明しなさい。

## 9 履歴

- 2023-04-28: 誤植の修正。プログラムのファイル名の変更（walrus → walras）
- 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。