応用一般均衡分析入門
第 8 章：関数形とカリブレーション *

武田 史郎 †

Date: 2022/01/28,
Version 3.0

目次

---

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

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

1 導入

今回の内容について。

- CGE 分析はモデルを数値的に解く分析であるので、当然であるが、**モデルに現れる全ての「関数の関数形」、及びそこに含まれる「パラメータの値」を特定化する必要がある。**
- 第 8 章ではまず CGE 分析における関数形の特定化について説明する。
- さらに、関数内のパラメータの値を特定化する方法である**「カリブレーション(calibration)」**について説明する。

[注]

「カリブレーション(calibration)」という用語(手法)は経済学の様々な分野で利用されているが、時に応じて違う意味で使われている。**ここでのカリブレーションはあくまで CGE 分析におけるカリブレーションである**ことに注意して欲しい[1]。

2 関数形の特定化

CGE モデルには生産関数、効用関数等、様々な関数が含まれる。理論分析では特定化しない、一般的な形で関数を表現して分析を進めることも多いが、数値的なシミュレーションをおこなうにはこれらの関数を特定化する必要がある。そこでどのような関数形に特定化するかが問題になる。

2.1 CGE 分析における関数形

結論から言うと、多くの CGE 分析では生産関数や効用関数に**「CES(constant elasticity of substitution)関数」**、あるいは、その特殊ケースである Cobb-Douglas 関数や Leontief 関数を利用している。CES 関数を使うことが多いのは、1) 単純で扱いやすい、2) 含まれるパラメータが少なく、パラメータ特定化が比較的簡単であるという理由が大きいと考えられる。

1) 例えば、マクロ経済学でも「カリブレーション」という手法が多用されているようであるが、この文書ではマクロ経済学のことについては何も扱っていない。

ただ、CES 関数には当然欠点もある。CES 関数は相似拡大的（homothetic）な関数である。効用関数に相似拡大的な関数を使うと需要の所得弾力性は常に 1 となってしまう。これはエンゲル曲線の関係（所得が上昇するにつれ、支出に占める食費のシェアが低下していく現象）を表現できないということを意味する。エンゲル曲線の関係は幅広く観察される現象であるので、それを表現できないことは大きな欠点である。

第二に、定義上、CES 関数では任意の二つの財の間の代替の弾力性が常に一定となってしまう。これは技術や選好について先験的に強い制限を置くことを意味する。ただ、財の間の代替の大きさについては、CES 関数を多段階にすることである程度柔軟に変更することはできる（ただし、それが常に一定ということは多段階にしても変わらないが)。これについては次節で説明する。

第 7 章まで関数の例として CES 関数を利用してきた。今後も関数形としては基本的に CES 関数を前提として話を進めていく。

2.2 多段階（入れ子型）の CES 関数

以下のような生産関数を考えよう。

$$y = f(\mathbf{x}) = \left[\sum_i \beta_i (x_i)^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}}$$

これは第 1 章で最初に例として考えた CES の生産関数である。もし、このような生産関数を仮定したとすると、任意の二つの投入物の間の代替の弾力性が常に σ に等しいということになる。これは、例えば、投入物として、原材料、エネルギー、サービス、資本、労働があったときに、原材料と労働の間の代替の程度が、エネルギーと資本の代替の程度に等しいというようなことを意味する。このように全く性質が異なると考えられる投入物の間の代替の弾力性が等しいと仮定することは、非常に強い制約となる。

そこでよく行なわれるのが、関数を多段階（入れ子型）にすることである。例えば、次のような生産関数があったとする。

$$y = h(z_{11}, \cdots, z_{1m}, z_{21}, \cdots, z_{2m}, \cdots, z_{n1}, \cdots, z_{nm})$$

関数 h は投入物 $\{z_{ij}\}$ の関数である。これを例えば二段階の CES 関数と仮定するとは、次のように f と g_i という二つの CES 関数を使って h を表すことである。

$$\begin{aligned} y &= h(z_{11}, \cdots, z_{1m}, z_{21}, \cdots, z_{2m}, \cdots, z_{n1}, \cdots, z_{nm}) \\ &= f(g_1(z_{11}, \cdots, z_{1m}), g_2(z_{21}, \cdots, z_{2m}), \cdots, g_n(z_{n1}, \cdots, z_{nm})) \end{aligned}$$

つまり、投入物をグループに分割し、一旦グループ毎に CES 関数により統合したものを、さらに CES 関数で統合するという形である。具体的には次のような表現になる。

$$y = f(x_1, \cdots, x_n) = \left[\sum_i \beta_i (x_i)^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}} \tag{1}$$

$$x_i = g_i(z_{1i}, \cdots, z_{mi}) = \left[\sum_j \delta_{ij} (z_{ij})^{\frac{\sigma_i^x-1}{\sigma_i^x}} \right]^{\frac{\sigma_i^x}{\sigma_i^x-1}} \tag{2}$$

この例では二段階としているが二段階には限らない。実際の CGE 分析ではもっと段階を多くしていることも多い。この二段階の生産関数においては、同じグループ内に属する投入物 x_{ij} と x_{il} の間の代替の弾力性は σ_i^x であるのに対し、違うグループに属する投入物 x_{ij} と x_{kl} の代替の弾力性は σ となる。このように CES 関数であっても多段階の関数形を仮定することで投入物間の代替の弾力性を変更することができる。

CGE 分析では、生産関数にしろ効用関数にしろ一段階の CES 関数を仮定することは少なく、多段階を仮定しているものがほとんどだと言ってよい。本稿でも、ここまで生産関数に二段階の CES 関数を利用してきた。多様な多段階の CES 関数を利用している例としては MIT の EPPA モデルがある。実際にどのような関数形が利用されているかは Chen et al. (2015) を参照して欲しい。

2.3 CES 関数以外の関数

CGE 分析では CES 関数の利用が多いが、CES 関数以外を利用しているモデルもある。CES 関数以外に利用されるものとしては例えば次の 3 つがある。

- CDE（constant difference of elasticities）関数
- Stone-Geary 効用関数
- Translog 関数

GTAP モデル (Hertel, 1999) では、効用関数（の一部）に CDE 関数（constant difference of elasticities function, Hertel et al. 1992）を利用している。

効用関数が CDE 関数であるとき、支出関数 $e(\mathbf{p},u)$ は次のように implicit に定義される。

$$\sum_i \gamma_i^{\sigma_i} \left[\frac{p_i}{e(\mathbf{p},u)} \right]^{1-\sigma_i} u^{(1-\sigma_i)\beta_i} = 1$$

$\sigma_i = \sigma$、$\beta_i = 1$ のときに、普通の CES 関数となる。

CDE 関数は非相似拡大的な関数であり、CES 関数では表現できないエンゲル曲線の関係を表現することができる。ただし、パラメータの数が多く、特定化の作業の難易度が比較的高いこともあり、GTAP モデル以外での採用はほとんどない。

2 つ目の Stone-Geary 関数は効用関数によく利用される。具体的には次のような関数である。

$$u = g(d_1, \cdots, d_n) = \phi \prod_i (d_i - \gamma_i)^{\alpha_i}$$

見た通り、形式としては Cobb-Douglas 関数に似ている。ただし、消費量にあたる変数から γ_i という値が差し引かれていることが普通の Cobb-Douglas 関数との違いである[2)]。この γ_i **は消費者が最低限消費しなければならない消費量**を表すパラメータと解釈できる。

Stone-Geary 関数は非相似拡大的な関数であり、エンゲル曲線の関係を考慮できる。Cobb-Douglas 関数と似ていて単純で、しかも追加的なパラメータは γ_i だけということから、非常

2) Stone-Geary 効用関数の無差別曲線は Cobb-Douglas 効用関数の無差別曲線の原点を $\{\gamma_1, \cdots, \gamma_n\}$ にシフトさせた形状になる。

によく利用される関数である。例えば、CGE 分析では次のモデルで Sotne-Geary 効用関数が利用されている。

- OECD の ENV-Linkages モデル (Chateau et al., 2014)
- GEM-E3 モデル[3)]
- World Bank の LINKAGE モデル (van der Mensbrugghe, 2005)
- IFPRI モデル (Lofgren et al., 2002)[4)]
- PEP モデル (Robichaud et al., 2012)[5)]

最初の 2 つは温暖化対策分析用のモデル、3 つ目は貿易政策の分析に使われているモデル、最後の 2 つは基本的な CGE モデルとして参照されるモデルである。

以上の 2 つに加え、translog 関数のような「柔軟な関数（flexible function）」を利用する場合もある。例えば、米国 EPA に利用されている IGEM モデル[6)]は関数形に translog 関数を利用している。translog 関数は自由度が高く、多様な技術、選好を表すことができるが、パラメータの数が非常に多く、特定化が難しいということもあって、CGE モデルでの利用は非常に少ない。

3 CGE 分析の仮定

3.1 基準均衡

関数形として CES 関数を利用することが多いということを説明した。次に問題になるのは、それではその CES 関数のパラメータをどう特定化するかという話であるが、その準備として、CGE 分析の基本的なアプローチについて説明しておく。一口にシミュレーションといっても様々なアプローチがある。一つの典型的なアプローチは

1) モデルを作成する
2) モデルのパラメータや外生変数を特定化する
3) モデルを解いて、未知の内生変数の値を求める

という手順に従うものであろう。実際、このようにモデルの解を求めるという作業を第 5 章と第 6 章でおこなった。第 5 章と第 6 章ではモデルを記述した後、生産関数、効用関数のパラメータの値（具体的には、ウェイトパラメータと代替の弾力性）を設定し、さらにモデルの外生変数である生産要素賦存量（や税率）を決めて、モデルを解いた。その結果、生産量、消費量、財価格、生産要素価格等の内生変数の値を求めることができた。このアプローチでは**内生変数の値というのはモデルを解いて求めるべきもので、当然それは事前には未知の値**である。

一方、CGE 分析は異なったアプローチをとる。CGE 分析では**「ベンチマークデータ（SAM）の**

3) https://ec.europa.eu/jrc/en/gem-e3 [2022-01-20 アクセス]
4) https://www.ifpri.org/publication/standard-computable-general-equilibrium-cge-model-gams-0 [2022-01-20 アクセス]
5) https://www.pep-net.org/pep-standard-cge-models [2022-01-20 アクセス]
6) http://www.igem.insightworks.com/docs/ [2022-01-16 アクセス]

下でモデルが均衡状態にある（その均衡を基準均衡 “benchmark equilibrium” と呼ぶ）」という前提を置き、その基準均衡が政策（や政策以外の）ショックの影響によって、どのような均衡に変化するかを分析する。「ベンチマークデータの下でモデルが均衡状態にある」ということは、ベンチマークデータにおける価格や数量の値が均衡条件を満たしているということであり、それは最初の段階でのモデルの内生変数の値が既知であるという状態から始めるということを意味する。言い換えれば、**モデルを解いて未知の変数を求めるというのではなく、既知の値にモデルの解が等しくなっている状況から始めるということ**である。

以上のように、ベンチマークデータの下でモデルが均衡状態にあるという前提で分析を始めることが CGE 分析の大きな特徴である。この前提を利用して一部のパラメータの値を決めるという方法がとられる。それがカリブレーションである。

3.2 注: 理論的な一般均衡分析との違い

経済学における一般均衡分析というと、Gérard Debreu や Kenneth J. Arrow を代表とする著名な経済学者達が取り組んだ理論的な一般均衡分析を思い浮べる人が多いかもしれない。それらの理論的な一般均衡分析での中心的な分析テーマは**「均衡の存在証明」**、つまり**どのような条件の下で一般均衡が存在するかを明らかにすること**だったと言えるだろう。CGE 分析でもそのようなテーマが扱われていると考える人もいるかもしれないが、実際には、両者の分析アプローチ、考え方は大きく異なっている。

上で説明したように、CGE 分析ではそもそも最初の時点で均衡状態にあるという前提で分析がスタートする。均衡状態にあるのだから、少なくとも最初の時点での均衡は存在するし、しかもそれは既知である。このようなアプローチをとることから、均衡が存在するかどうかという理論的な一般均衡分析の視点で考えることは CGE 分析ではほとんどないと言ってよい

ただし、CGE 分析でも、均衡の存在についての問題が全く出てこないわけではない。最初の時点では存在するとしても、何らかのショックを与えた後では均衡が存在しなくなる場合もありうる。例えば、元々はなかった新しい政策を導入したとすると、それによりモデルの解が存在しなくなるということもありうる。しかし、筆者の経験から言うと、不適切な方法でショックを加えない限り、CGE 分析において解の存在の問題が生じることは少ない。

3.3 注: 以前の CGE 分析との違い

第 8 章のトピックとは直接関係のない話であるが、理論的な一般均衡分析との違いということに触れたので、ついでに初期の CGE 分析と最近の CGE 分析の違いにも触れておこう。CGE 分析（応用一般均衡分析）が始まったのは Johansen (1960) からと言われているが[7]、70 年代から 80 年代の発展段階における研究者としては特に John B. Shoven と John Whalley の二人が有名である[8]。彼等の研究では CGE モデルを解くのに Herbert Scarf により考案されたアルゴリズム（Scarf アルゴリズム）が利用されている。そして、この二人によって書かれた CGE 分析についてのテキスト (Shoven and Whalley, 1992) でも Scarf アルゴリズムが説明されている。また、彼等と

7) Dixon and Jorgenson (2013)。
8) 研究書としては、Ballard et al. (1985) が有名である。

同様のアプローチを使った分析をしている市岡 (1991) でも Scarf アルゴリズムについて説明されている。このように、昔の CGE 分析についての文献では数値計算についてのアルゴリズムが解説されていることが多い。

このような昔の書籍を読めば、CGE 分析をおこなうのにはそういった数値計算のアルゴリズムについて習熟する必要があるように思えるかもしれない。しかし、現在 CGE 分析をおこなうときには自分でアルゴリズムを考えたり、実装したりというような必要はほとんどない。CGE モデルを解くということは数学的には非線形の連立方程式を解くという問題（あるいは、非線形の最適化という問題）であり、それは別に一般均衡分析（経済学）に固有の問題ではなく、自然科学も含めて様々な分野で扱われている問題である。そして、現在はそのような問題を解くためのソフトウェアが多数存在しているので、それらを用いれば CGE モデルを解くことができる。実際、この文書では GAMS という数値計算ソフトを利用して CGE モデルを解いているが、GAMS は CGE モデル専用のソフトウェアでもなんでもなく[9)]、様々なタイプの数値計算問題を解くための汎用のソフトウェアである。もちろんソフトウェアによって性能や使い勝手の差があり、その選択は重要であるが、いずれにせよ自分でアルゴリズムを考えたり、実装したりする必要はほとんどないと言ってよい。

4 カリブレーション

4.1 カリブレーションとは？

それでは次に関数形内のパラメータの特定化の問題を考える。以下では第 5 章で利用したモデルを例にとる。

$$c_i = \left[\sum_j (\alpha^x_{ji})^{\sigma_i} (p_j)^{1-\sigma_i} + (\alpha^v_i)^{\sigma_i} (p^{va}_i)^{1-\sigma_i} \right]^{\frac{1}{1-\sigma_i}} \qquad \{c_i\}_{i=1,\cdots,n} \tag{3}$$

$$c^{va}_i = \left[\sum_f (\beta^v_{fi})^{\sigma^v_i} \left[(1+t^F_{fi}) p^F_f \right]^{1-\sigma^v_i} \right]^{\frac{1}{1-\sigma^v_i}} \qquad \{c^{va}_i\}_{i=1,\cdots,n} \tag{4}$$

$$c_i - p_i = 0 \qquad \{y_i\}_{i=1,\cdots,n} \tag{5}$$

$$c^{va}_i - p^{va}_i = 0 \qquad \{v^a_i\}_{i=1,\cdots,n} \tag{6}$$

$$a^x_{ji} = \left[\frac{\alpha^x_{ji} c_i}{p_j} \right]^{\sigma_i} \qquad \{a^x_{ji}\}_{i,j=1,\cdots,n} \tag{7}$$

$$a^v_i = \left[\frac{\alpha^v_i c_i}{p^{va}_i} \right]^{\sigma_i} \qquad \{a^v_i\}_{i=1,\cdots,n} \tag{8}$$

$$x_{ij} = a^x_{ij} y_j \qquad \{x_{ij}\}_{i,j=1,\cdots,n} \tag{9}$$

$$a^F_{fi} = \left[\frac{\beta^v_{fi} c^{va}_i}{(1+t^F_{fi}) p^F_f} \right]^{\sigma^v_i} \qquad \{a^F_{fi}\}_{i=1,\cdots,n,f=1,\cdots,m} \tag{10}$$

9) MPSGE という CGE モデル用の機能も提供されているが、基本的には汎用の数値計算ソフトである。

$$v_{fi}=a^F_{fi}v^a_i \qquad \{v_{fi}\}_{f=1,\cdots,m,i=1,\cdots,n} \tag{11}$$

$$e=\left[\sum_j(\gamma_j)^{\sigma^c}\left[(1+t^C_j)p_j\right]^{1-\sigma^c}\right]^{\frac{1}{1-\sigma^c}}u \qquad \{e\} \tag{12}$$

$$d_i=\left[\frac{\gamma_i(e/u)}{(1+t^C_i)p_i}\right]^{\sigma^c}u \qquad \{d_i\}_{i=1,\cdots,n} \tag{13}$$

$$y_i=\sum_j a^x_{ij}y_j+d_i \qquad \{p_i\}_{i=1,\cdots,n} \tag{14}$$

$$v^a_i=a^v_i y_i \qquad \{p^{va}_i\}_{i=1,\cdots,n} \tag{15}$$

$$\bar{v}_f=\sum_i a^F_{fi}v^a_i \qquad \{p^F_f\}_{f=1,\cdots,m} \tag{16}$$

$$m=\sum_f p^F_f\bar{v}_f+\sum_i t^C_i p_i d_i+\sum_{i,f}t^F_{fi}p^F_f v_{fi} \qquad \{m\} \tag{17}$$

$$e=m \qquad \{u\} \tag{18}$$

これは第 5 章のモデルの双対アプローチによる均衡条件の表現を少し書き換えたものである。単位費用を表す変数 c_i と c^{va}_i、支出を表す変数 e を導入して第 5 章の均衡条件を書き直している。

このモデルの関数にはパラメータとして以下に挙げるものが含まれている。

- 生産関数内の α^x_{ji}、α^v_i、σ_i、β^v_{fi}、σ^v_i
- 効用関数内の γ_i、σ^c

ここでは CES 関数を利用しているので、関数に含まれるパラメータは結局

- ウェイトパラメータ
- 代替の弾力性

の二種類である。シミュレーションをおこなうにはこれらのパラメータを特定化する必要がある。パラメータの特定化には様々なアプローチがありうるが、CGE 分析では、1) **代替の弾力性を何らかの情報を基に外生的に設定**し、2) **ウェイトパラメータをカリブレート**するという方法がとられることが多い。

「カリブレート（calibrate）」、あるいは**「カリブレーション（calibration）」**という用語は経済学の様々な分野で利用されており、しかも利用される状況によって異なった意味を付与されていることが多い。このため、どのような意味で利用しているのか注意しなければならないが、CGE 分析におけるカリブレーションは、

> **定義** 1: ベンチマークデータ（SAM）の下でモデルが均衡するようにモデル内のパラメータの値を設定すること

を意味することが多い。あるいは、もう少し広く

定義 2: 外生的に与えるデータ・数値とモデルから導かれる値が整合的になるようにモデル内のパラメータの値を設定すること

というような意味で用いられることもある。

「ウェイトパラメータをカリブレートする」とは「定義 1」の意味である。つまり、ベンチマークデータの下でモデルが均衡するようにウェイトパラメータを決定するということである。

「定義 2」では、ベンチマークデータに限らず、外生的に与えるデータや数値にモデルから導かれるデータや数値が一致するようにパラメータを決定するという意味になる。上で、「代替の弾力性を外生的に設定」すると書いたが、「代替の弾力性をカリブレートする」場合もある。例えば、効用関数内の「消費と余暇の代替の弾力性」を、モデルにおける労働供給の賃金弾力性が外生的に与える賃金弾力性値に等しくなるようにカリブレートすることがある(これについては第 A1 章で説明している)。こちらの場合には、労働供給の賃金弾力性の値というベンチマークデータ(SAM)ではないデータを基にしているという意味で定義 1 には当てはまらないが、このような場合もカリブレーションと呼ぶことが多い。こちらのカリブレーションも含んだ定義が「定義 2」である。いずれにせよ、「カリブレーション」という用語は定義が曖昧で、状況によって異なる意味で使われることがあるので、注意が必要である。

以下では、**「定義 1 の方法でのウェイトパラメータのカリブレーション」**を主に考えていく。

4.2 カリブレーションの手順

それでは実際にウェイトパラメータをカリブレートする手順について説明しよう。例として、(7) 式、(8) 式の α_{ji}^x、α_i^v というウェイトパラメータを考えよう。均衡条件が満たされるということは企業はベンチマークデータの下で費用最小化行動をとっていることになる。ベンチマークデータにおける変数の値を全てバー付きの変数で表すとすると、これは次の式が満たされていることを意味する。

$$\bar{a}_{ji}^x = \left[\frac{\alpha_{ji}^x \bar{c}_i}{\bar{p}_j}\right]^{\sigma_i} \qquad \bar{a}_i^v = \left[\frac{\alpha_i^v \bar{c}_i}{\bar{p}_i^{va}}\right]^{\sigma_i}$$

これを書き換えると次式を得る。

$$\alpha_{ji}^x = \frac{\left(\bar{a}_{ji}^x\right)^{\frac{1}{\sigma_i}} \bar{p}_j}{\bar{c}_i} \qquad \alpha_i^v = \frac{\left(\bar{a}_i^v\right)^{\frac{1}{\sigma_i}} \bar{p}_i^{va}}{\bar{c}_i}$$

さらに、

$$\theta_{ji}^x \equiv \frac{\bar{p}_j \bar{a}_{ji}^x}{\bar{c}_i} = \frac{\bar{p}_j \bar{a}_{ji}^x}{\sum_l \bar{p}_l \bar{a}_{li}^x + \bar{p}_i^{va} \bar{a}_i^v} \qquad \theta_i^v \equiv \frac{\bar{p}_i^{va} \bar{a}_i^v}{\bar{c}_i} = \frac{\bar{p}_i^{va} \bar{a}_i^v}{\sum_l \bar{p}_l \bar{a}_{li}^x + \bar{p}_i^{va} \bar{a}_i^v}$$

と定義すると、次式に書き換えることができる。

$$\alpha_{ji}^x = \theta_{ji}^x \left(\bar{a}_{ji}^x\right)^{\frac{1-\sigma_i}{\sigma_i}} \qquad \alpha_i^v = \theta_i^v \left(\bar{a}_i^v\right)^{\frac{1-\sigma_i}{\sigma_i}} \tag{19}$$

(19) 式の右辺にはベンチマークデータに含まれる変数、及び代替の弾力性しか現われない。よって、代替の弾力性の値を決めれば、後はベンチマークデータからウェイトパラメータ α_{ji}^x、α_i^v の値を求めることができる。これが「ウェイトパラメータのカリブレーション」である。

他のウェイトパラメータについても同様にカリブレートできる。

$$\alpha^x_{ji} = \frac{\left(\bar{a}^x_{ji}\right)^{\frac{1}{\sigma_i}} \bar{p}_j}{\bar{c}_i} = \theta^x_{ji} \left(\bar{a}^x_{ji}\right)^{\frac{1-\sigma_i}{\sigma_i}} \tag{20}$$

$$\alpha^v_i = \frac{\left(\bar{a}^v_i\right)^{\frac{1}{\sigma_i}} \bar{p}^{va}_i}{\bar{c}_i} = \theta^v_i \left(\bar{a}^v_i\right)^{\frac{1-\sigma_i}{\sigma_i}} \tag{21}$$

$$\beta^v_{fi} = \frac{\left(\bar{a}^F_{fi}\right)^{\frac{1}{\sigma^v_i}} \bar{\tilde{p}}^F_{fi}}{\bar{c}^{va}_i} = \theta^F_{fi} \left(\bar{a}^F_{fi}\right)^{\frac{1-\sigma^v_i}{\sigma^v_i}} \tag{22}$$

$$\gamma_i = \frac{\left(\bar{d}_i/\bar{u}\right)^{\frac{1}{\sigma^c}} \bar{\tilde{p}}^c_i \bar{u}}{\bar{e}} = \theta^c_i \left(\bar{d}_i/\bar{u}\right)^{\frac{1-\sigma^c}{\sigma^c}} \tag{23}$$

ただし、$\tilde{p}^F_{fi} \equiv (1+t^F_{fi})p^F_f$、つまり $\tilde{p}^F_{fi}$ を生産者が直面する生産要素価格とし、そのベンチマークでの値を $\bar{\tilde{p}}^F_{fi} = (1+\bar{t}^F_{fi})\bar{p}^F_f$ とする。同様に、$\tilde{p}^c_i \equiv (1+t^c_i)p_i$、つまり $\tilde{p}^c_i$ を消費者が直面する財の価格とし、そのベンチマークでの値を $\bar{\tilde{p}}^c_i = (1+\bar{t}^c_i)\bar{p}_i$ とする。このように tilde 付きの価格を**「agent 価格」**を表すものとする。「agent 価格」とは経済主体が直面する価格のことであり、上の $\tilde{p}^F_{fi}$ や $\tilde{p}^C_i$ のように税込み（あるいは、補助金込み）の価格のことである。

さらに、θ は次のように定義される。

$$\theta^x_{ji} \equiv \frac{\bar{p}_j \bar{a}^x_{ji}}{\bar{c}_i} = \frac{\bar{p}_j \bar{a}^x_{ji}}{\sum_l \bar{p}_l \bar{a}^x_{li} + \bar{p}^{va}_i \bar{a}^v_i}$$

$$\theta^v_i \equiv \frac{\bar{p}^{va}_i \bar{a}^v_i}{\bar{c}_i} = \frac{\bar{p}^{va}_i \bar{a}^v_i}{\sum_l \bar{p}_l \bar{a}^x_{li} + \bar{p}^{va}_i \bar{a}^v_i}$$

$$\theta^F_{fi} \equiv \frac{\bar{\tilde{p}}^F_f \bar{a}^F_{fi}}{\bar{c}^{va}_i} = \frac{\bar{\tilde{p}}^F_f \bar{a}^F_{fi}}{\sum_k \bar{\tilde{p}}^F_k \bar{a}^F_{ki}}$$

$$\theta^c_i \equiv \frac{\bar{\tilde{p}}^c_i \bar{d}_i}{\bar{e}} = \frac{\bar{\tilde{p}}^c_i \bar{d}_i}{\sum_j \bar{\tilde{p}}^c_j \bar{d}_j}$$

ベンチマークにおける $\bar{p}_i$、$\bar{p}^{va}_i$、$\bar{p}^F_f$、$\bar{a}^x_{ji}$、$\bar{a}^v_i$、$\bar{a}^F_{fi}$、$\bar{m}$、$\bar{u}$、$\bar{t}^F_{fi}$、$\bar{t}^C_i$ の値と、代替の弾力性の値、σ_i、σ^v_i、σ^c を設定することで、α^x_{ji}、α^v_i、β^v_{fi}、γ_i の値が計算できる。「ベンチマークデータの下でモデルが均衡するようにウェイトパラメータを決定する」とは以上のような意味である。

4.3　カリブレーションの例

それでは実際にデータを用いてウェイトパラメータのカリブレーションをしてみよう。ベンチマークデータとしては、表 1 の SAM を利用する。これは第 7 章で例として出した SAM である。

表 1：SAM の例

	農業	製造業	サービス	労働	資本	家計	行和
農業	30	10	30			70	140
製造業	10	50	20			220	300
サービス	20	40	20			70	150
労働	50	80	50				180
資本	30	120	30				180
家計				180	180		360
列和	140	300	150	180	180	360	

SAM の項目では部門・財は「農業」、「製造業」、「サービス」、生産要素は「労動」、「資本」となっているが、以下では部門・財を $i = \{\mathtt{AGR}, \mathtt{MAN}, \mathtt{SER}\}$、生産要素を $f = \{\mathtt{LAB}, \mathtt{CAP}\}$ と表現する。

4.3.1 Harberger Convention

カリブレーションをおこなうにはベンチマークの価格や数量を与えてやる必要がある。それにはSAM のデータを使うのであるが、ここで一つ問題がある。SAM のデータは基本的に「金額（＝価格 × 数量)」であり、カリブレーションに必要な「価格」と「数量」のデータに分解されていないという問題である。これに対処するために、通常 “Harberger convention” という想定が置かれる。“Harberger convention” **とはベンチマークにおける価格を 1 と置く**という想定である。こうすると「金額＝ 1 × 数量＝数量」ということになり、元々は金額である SAM のデータをそのまま数量のデータと解釈できる。

ベンチマーク価格をどのような数値に設定するかによって、当然ベンチマークの数量が変わってくる。例えば、ある財の生産額が 100 としたとき、財の価格を 1 と置けば生産量は 100 となるが、2 と置けば 50 となる。単純に価格を 1 と置いてしまって問題ないのかと疑問が湧くであろうが、結論だけ述べると、**「分析において変数の変化率のみを見る場合には、ベンチマークの価格をどのような値に置いても分析結果は変わらない」**という性質が成り立つので問題ない[10)]。どのような値に置いてもよいのなら、1 に規準化するのが単純で最もわかりやすい。このため 1 に置くのが普通である。

ベンチマーク価格を 1 に置くことの可否については、第 7.2 節でもう一度詳しくチェックする。また、ベンチマーク価格を 1 に規準化すると言っても、全ての価格を 1 に設定できないケースもある。例えば、企業が支払う賃金を w、労働者が受け取る賃金を $\tilde{w}$、労働所得に対する税率を t^L とし、この 3 つの間に $\tilde{w} = (1 - t^L)w$ という関係が成り立っているとする。この場合、もし企業が支払う賃金 w を 1 に規準化すると、労働者にとっての賃金 $\tilde{w}$ は自動的に 1 ではなくなることになる。このように 1 と規準化できない価格も出てくる場合があるが、可能なときには基本的に 1 と規準化すればよい。

[注]

SAM データは「金額データ」であり、「価格」と「数量」に分割されていないと書いたが、これは単にデータの不備というわけではなく、そもそも原理的に「価格」と「数量」に分割することが難しいことが多い。例えば、自動車産業の生産額を考えよう。自動車と言っても、乗用車もあればトラック、バスのような大型車もある。さらに、乗用車と言ってもその種類は多様で、価格にも大きな幅がある。多様な種類の自動車から構成される生産額を価格と数量に分割しようとしても難しいことは明らかである。仮に財の分類を非常に細かくすれば価格、数量の分割はしやすくなるかもしれないが、そうすると今度は CGE 分析で用いるには細かすぎる分類になってしまう。自動車のようなモノでも難しいが、サービスという数量で表現しにくいものの場合には一層価格、数量の分割は難しくなる。

10) ただし、これは第 2 章で扱った「価格についての 0 次同次性」が成り立つモデルでの話である。

4.3.2 カリブレーションのプログラム

calibration_example.gms がカリブレーションをおこなっているプログラムである。基本的には第 5 章における ge_sample_dual.gms のコードと共通部分が多いが、モデルの表現を若干変更している。以下では、この calibration_example.gms のコードを解説していく。

```
*       --------------------------------------------------------------
*       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;

*       --------------------------------------------------------------
*       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。
+ これをカリブレートする。
$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
;
*       --------------------------------------------------------------
*       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。
+ Chap 5 のときと同じ値を設定。
$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;
```

まず、ここは ge_sample_dual.gms と全く同じである。最初に財・部門、生産要素を表す集合 i と f を定義している。 その次に、ウェイトパラメータを宣言している。ge_sample_dual.gms で

はこのウェイトパラメータの値は外生的に与えていた。今回はこの 3 つのウェイトパラメータの値をカリブレーションによって求めるという作業をおこなう。

ウェイトパラメータの次に代替の弾力性を表すパラメータを宣言し、その値を設定している。値には ge_sample_dual.gms と同様に代替の代替の弾力性値は外生的に設定する。ここでは 3 つの代替の弾力がどれも 0.5 であると仮定している。

```
*       ------------------------------------------------------------
*       ベンチマークデータ
$ontext
以下を SAM として利用する。
$offtext

table   SAM     SAM data (benchmark data)
              agr       man       ser       lab       cap        hh
agr            30        10        30         0         0        70
man            10        50        20         0         0       220
ser            20        40        20         0         0        70
lab            50        80        50         0         0         0
cap            30       120        30         0         0         0
hh              0         0         0       180       180         0
;
display SAM;

set     row     / agr, man, ser, lab, cap, hh /;
alias(col,row), (row, roww);
```

ここでは SAM を定義している。SAM は二次元データであるので、table 命令を用いて定義している。中身は 1 の SAM そのままである。また、行・列の項目を表す集合も一緒に定義している。

```
$ontext
以下の 0 付きのパラメータはベンチマークの変数の値を表すパラメータ
$offtext
parameters
    p0(i)         財の価格
    p_va0(i)      合成生産要素の価格
    p_f0(f)       生産要素の価格

    y0(i)         生産量
    vf0(f,i)      生産要素需要
    x0(j,i)       投入需要
    v_a0(i)       合成生産要素
    a_x0(j,i)     単位投入需要
    a_v0(i)       単位合成生産要素需要
    a_f0(f,i)     単位生産要素需要
```

```
    d0(i)       消費需要
    m0          所得
    u0          効用水準

    c0(i)       生産の単位費用
    c_va0(i)    合成生産要素生産の単位費用
    e0          支出
;
```

ここで変数のベンチマークの値を表すパラメータを宣言している。後に内生変数を宣言する部分がでてくるが、ここではその内生変数の記号に「0」を付けることでベンチマークの値を表すものとしている。以下、これらのパラメータに値を代入していく。

```
 *      Harberger convention（全ての価格を 1 に規準化）
p0(i) = 1;
p_va0(i) = 1;
p_f0(f) = 1;
```

ここでベンチマークの価格を設定している。既に述べたように、このような場合 Harberger convention が使われる。Harberger convention に従いベンチマークにおいて全ての価格は 1 と設定している。

```
*       生産量＝生産額/価格
y0(i) = sum(col, SAM(col,i)) / p0(i);
```

以下、ベンチマークの数量を求めていく。まず、ここではベンチマークの生産量を求めている。生産額は SAM から入手できるので、それを価格で割ってやれば生産量になる。価格には 1 を設定しているので、`p0(i)` で割るという操作は実質的な意味は持たないが、計算方法の考え方を示すために `p0(i)` で割るという表現を省略せずに入れている。後で、ベンチマークの価格を 1 以外の価格に設定するというケースも考えるので、そのときには価格で割るという表現が実質的な意味も持ってくることになる。

```
*       投入需要量＝需要額/価格
x0(j,i) = SAM(j,i) / p0(j);
```

ここでは中間投入需要量を求めている。方法は先程と同じで、SAM の金額データを価格で割る

という形で求めている。

```
*       合成生産要素需要量＝需要額/価格
v_a0(i) = sum(f, SAM(f,i)) / p_va0(i);

*       生産要素需要量＝需要額/価格
vf0(f,i) = SAM(f,i) / p_f0(f);
```

合成生産要素需要量、生産要素需要量についても同じように、「金額/価格」という関係により求めている。

```
*       単位投入需要量＝投入需要量/生産量
a_x0(j,i) = x0(j,i) / y0(i);
```

ここではベンチマークの単位中間投入需要量（つまり、生産量一単位あたりの中間投入量）を求めている。単位中間投入需要量の定義より、それは「総」中間投入需要量を生産量で割ったものである。

```
*       単位合成生産要素需要量＝合成生産要素需要量/生産量
a_v0(i) = v_a0(i) / y0(i);

*       単位生産要素需要量＝生産要素需要量/生産量
a_f0(f,i) = vf0(f,i) / v_a0(i);
```

この二つも単位需要量である。総需要量を生産量で割ることで求めることができる。

```
*       消費需要量＝消費需要額/価格
d0(i) = SAM(i,"hh") / p0(i);
```

これはベンチマークの消費需要量である。生産サイドの需要量と同様に SAM の金額データを価格で割って求めている。

```
*      所得額
m0 = sum(f, SAM("hh",f));
```

これはベンチマークの所得額である。これはそのまま SAM のデータから求められる。

```
*       効用水準
u0 = m0;
```

ここでベンチマークの効用水準を設定している。通常、CGE 分析では効用水準は変化率の観点からしか考えない。つまり、**「効用水準が何%変化するか」**ということは考えるが、**「効用水準がいくつか」**ということは考えない[11)]。変化率の観点しか考えないので、価格のときと同様に、ベンチマークの水準を適当な値に規準化してよい。ここではベンチマークの効用水準を所得の水準に等しいと規準化している。別の値に設定してもよいのであるが、こうするとベンチマークにおける効用の価格（効用一単位を得るのに必要な支出）を 1 と設定することと同じになるので、他の価格を 1 と設定しているのと整合的になる。

```
*       生産の単位費用
c0(i) = p0(i);

*       合成生産要素生産の単位費用
c_va0(i) = p_va0(i);
```

ここでは単位費用を設定している。均衡では単位費用＝価格が成立しているので、既に設定済みの価格の値で決まってくる。価格には 1 を設定しているので、ベンチマークの単位費用も 1 となる。

```
*       支出
e0 = m0;
```

ここではベンチマークの支出額を設定している。支出額＝所得が成り立っていなければならないので、既に設定済みの所得額に等しい。

```
*       ---------------------------------------------------------------
*       カリブレーション

alpha_x(j,i) = (a_x0(j,i))**(1/sig(i)) * p0(j) / c0(i);
```

11) （ミクロ）経済学のテキストでは効用は通常「序数的」な意味しか持たない、つまり効用の水準の大小関係は意味があるが、いくつだとか、何倍だとかというような情報には意味がないというように説明されていることがある。CGE 分析でも効用がいくつかということには意味があるとは考えられないが、何%変化するかという値には意味があるとみなされることが多い。この意味で CGE 分析では効用をある程度「基数的」な意味で解釈すると言える。

```
alpha_v(i) = (a_v0(i))**(1/sig(i)) * p_va0(i) / c0(i);

beta_v(f,i) = (a_f0(f,i))**(1/sig_v(i)) * p_f0(f) / c_va0(i);

gamma(i) = (d0(i)/u0)**(1/sig_c) * p0(i) * u0 / e0;

option alpha_x:8, alpha_v:8, beta_v:8, gamma:8;
display alpha_x, alpha_v, beta_v, gamma;
```

ここでウェイトパラメータをカリブレートしている。カリブレートに利用している式は (20) 式から (23) 式までと全く同じである。

```
option alpha_x:8, alpha_v:8, beta_v:8, gamma:8;
```

という命令はパラメータの中身を display で表示する際に、小数点以下何桁まで表示するかを設定している。このコードでは 4 つのパラメータについて小数点以下 8 桁まで表示せよということになる。パラメータ全体について小数点以下 8 桁まで表示させるようにするには

```
option decimals = 8;
```

とすればよい。

```
*       --------------------------------------------------------------
*       変数の宣言
variables
    c(i)        生産の単位費用
    c_va(i)     合成生産要素生産の単位費用
    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
    e           支出
    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
```

```
;
*       -------------------------------------------------------------
*       式の宣言
equations
    e_c(i)       生産の単位費用
    e_c_va(i)    合成生産要素生産の単位費用
    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_e          支出関数
    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
;
```

ここでは変数と式を宣言している。基本的に (3) 式〜(18) 式の変数に対応している。

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

*       生産の単位費用
e_c(i) .. c(i) =e=
          (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)));

*       合成生産要素生産の単位費用
e_c_va(i) .. c_va(i) =e=
             (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)));

*       生産における利潤最大化条件
e_y(i) .. c(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) .. c_va(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) * c(i)/p(j))**(sig(i));

*       単位合成生産要素需要
e_a_v(i) .. a_v(i) =e= (alpha_v(i) * c(i)/p_va(i))**(sig(i));

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

*       支出関数
e_e .. e =e=
       u * (sum(i,
           (gamma(i))**(sig_c) * ((1+t_c(i))*p(i))**(1-sig_c)))**(1/(1-sig_c));

*       消費需要
e_d(i) .. d(i) =e=
          u * (gamma(i)*(e/u)/((1+t_c(i))*p(i)))**(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 ..    e - m =e= 0;

*       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);
```

ここでは式を定義している。(3) 式〜(18) 式の定義そのままである。

```
*       --------------------------------------------------------------
*       変数の初期値
```

```
$ontext
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。
$offtext

c.l(i) = c0(i);
c_va.l(i) = c_va0(i);
y.l(i) = y0(i);
v_a.l(i) = v_a0(i);
a_x.l(j,i) = a_x0(j,i);
a_v.l(i) = a_v0(i);
a_f.l(f,i) = a_f0(f,i);
e.l = e0;
d.l(i) = d0(i);
x.l(j,i) = a_x0(j,i) * y0(i);
vf.l(f,i) = a_f0(f,i) * v_a0(i);
p.l(i) = 1;
p_va.l(i) = 1;
p_f.l(f) = 1;
u.l = u0;
m.l = m0;
```

ここで変数の初期値を設定している。ge_sample_dual.gms では変数の初期値には別の値を設定していたが、ここでは変数の初期値にその変数のベンチマークの値を設定している。そもそもベンチマークデータが均衡条件を満たすようにパラメータをカリブレートしたのであるから、「モデルの解＝ベンチマークの値」が成り立っていなければならない。

```
*       --------------------------------------------------------------
*       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;

$exit
```

ここでモデルを解いている。解いた結果は次の通りである。

```
---- VAR y  生産量

           LOWER          LEVEL          UPPER         MARGINAL

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

```
ser           .           150.0000        +INF            .

---- VAR v_a  合成生産要素

            LOWER          LEVEL          UPPER         MARGINAL

agr           .            80.0000        +INF            .
man           .           200.0000        +INF            .
ser           .            80.0000        +INF            .
```

これは calibration_example.lst における計算結果の部分である。生産量 y(i) と合成生産要素 v_a(i) の解がともにベンチマークの値に等しくなっているのが確認できる。他の変数についても全てベンチマーク値として設定した値に等しくなる。自分で確認して欲しい。「ベンチマークデータの下で均衡が成立している」という前提でパラメータがカリブレートされたのだから、「解＝ベンチマーク値」となっていなければならない。もし、そうなっていなければ、それはプログラム、モデル、データのどこかがおかしいということである。

以上のように、CGE 分析ではベンチマークデータの下で均衡状態にあるという想定を置くことによってウェイトパラメータをカリブレートするのである。

solve 命令の次にある$exit 命令をコメントアウトすれば、その後のシミューレションを解く。その後では、ge_sample_dual.gms と同様に、資本の賦存量を変化させるシミュレーション、及び、消費税を導入するシミュレーションをおこなっている。ge_sample_dual.gms の計算結果と同じ結果が導かれているかどうか確認して欲しい。

5 Calibrated Share Form (CSF) の CES 関数

第 4 節では CES 関数のウェイトパラメータをカリブレーションを考えた。ここでは、CES 関数の calibrated share form (以下、CSF) の解説をおこなう。

5.1 CSF の CES 関数とは

第 4 節では、カリブレーションによってウェイトパラメータの値を求め、それをウェイトパラメータに代入し、パラメータの特定化をおこないシミュレーションをおこなった。この方法では、ウェイトパラメータをカリブレートする部分のプログラムを別途に作成しなければならない。前節で利用した例ではカリブレートするパラメータは少なく単純であったが、シミュレーションにおいて様々な形式の CES 関数を同時に用いている場合には、これが非常に繁雑な作業となりうる。CSF は別途にウェイトパラメータのカリブレートするという作業をおこなわずに CES 関数を表現する方法である。

CSF とはウェイトパラメータに、そのカリブレートされた値を代入した形式のことである。再び α_{ji}^x、α_i^v の二タイプのウェイトパラメータを考えよう。これらは (20) 式、(21) 式でその値をカ

リブレートできる。こうしてカリブレートされた値を元の CES 関数に代入したものが CSF である。カリブレートされた値を代入してしまうので、α^x_{ji}、α^v_i というパラメータは消え、ベンチマークデータが直接関数に入ってくる形になる。

5.1.1 CSF の生産関数

実際に、CSF の CES 関数を求めてみよう。元の生産関数は

$$y_i = f^y_i(x_{1i}, \cdots, x_{ni}, v^a_i) = \left[\sum_j \alpha^x_{ji}(x_{ji})^{\frac{\sigma_i - 1}{\sigma_i}} + \alpha^v_i (v^a_i)^{\frac{\sigma_i - 1}{\sigma_i}} \right]^{\frac{\sigma_i}{\sigma_i - 1}}$$

であった。この生産関数の α^x_{ji}、α^v_i に (20) 式、(21) 式でカリブレートされた値を代入する。

$$\begin{aligned} y_i &= \left[\sum_j \left[\theta^x_{ji}(\bar{a}^x_{ji})^{\frac{1-\sigma_i}{\sigma_i}} \right] (x_{ji})^{\frac{\sigma_i - 1}{\sigma_i}} + \left[\theta^v_i (\bar{a}^v_i)^{\frac{1-\sigma_i}{\sigma_i}} \right] (v^a_i)^{\frac{\sigma_i - 1}{\sigma_i}} \right]^{\frac{\sigma_i}{\sigma_i - 1}} \\ &= \left[\sum_j \theta^x_{ji} \left(\frac{\bar{x}_{ji}}{\bar{y}_i} \right)^{\frac{1-\sigma_i}{\sigma_i}} (x_{ji})^{\frac{\sigma_i - 1}{\sigma_i}} + \theta^v_i \left(\frac{\bar{v}^a_i}{\bar{y}_i} \right)^{\frac{1-\sigma_i}{\sigma_i}} (v^a_i)^{\frac{\sigma_i - 1}{\sigma_i}} \right]^{\frac{\sigma_i}{\sigma_i - 1}} \\ &= \left[\left(\frac{1}{\bar{y}_i} \right)^{\frac{1-\sigma_i}{\sigma_i}} \sum_j \theta^x_{ji} \left(\frac{x_{ji}}{\bar{x}_{ji}} \right)^{\frac{\sigma_i - 1}{\sigma_i}} + \left(\frac{1}{\bar{y}_i} \right)^{\frac{1-\sigma_i}{\sigma_i}} \theta^v_i \left(\frac{v^a_i}{\bar{v}^a_i} \right)^{\frac{\sigma_i - 1}{\sigma_i}} \right]^{\frac{\sigma_i}{\sigma_i - 1}} \end{aligned}$$

以上より、次式を得る。

$$y_i = \bar{y}_i \left[\sum_j \theta^x_{ji} \left(\frac{x_{ji}}{\bar{x}_{ji}} \right)^{\frac{\sigma_i - 1}{\sigma_i}} + \theta^v_i \left(\frac{v^a_i}{\bar{v}^a_i} \right)^{\frac{\sigma_i - 1}{\sigma_i}} \right]^{\frac{\sigma_i}{\sigma_i - 1}}$$

これが CSF による CES 関数である。ウェイトパラメータが消去されているので、式に現れるのはベンチマークデータと代替の弾力性と通常の変数である。

同様に計算することで、v^a_i と効用関数についても CSF が導ける。

$$\begin{aligned} v^a_i &= \bar{v}^a_i \left[\sum_f \theta^F_{fi} \left(\frac{v_{fi}}{\bar{v}_{fi}} \right)^{\frac{\sigma^v_i - 1}{\sigma^v_i}} \right]^{\frac{\sigma^v_i}{\sigma^v_i - 1}} \\ u &= \bar{u} \left[\sum_i \theta^C_i \left(\frac{d_i}{\bar{d}_i} \right)^{\frac{\sigma^c - 1}{\sigma^c}} \right]^{\frac{\sigma^c}{\sigma^c - 1}} \end{aligned}$$

5.1.2 CSF の単位費用関数

次に、単位費用関数の CSF を求める。元々の部門 i の単位費用関数は次式であった。

$$c_i(\mathbf{p}, p^{va}_i) = \left[\sum_j (\alpha^x_{ji})^{\sigma_i} (p_j)^{1-\sigma_i} + (\alpha^v_i)^{\sigma_i} (p^{va}_i)^{1-\sigma_i} \right]^{\frac{1}{1-\sigma_i}}$$

これに α^x_{ji}、α^v_i のカリブレートした値を代入して書き換える。

$$
\begin{aligned}
c_i(\mathbf{p}, p^{va}_i) &= \left[\sum_j \left[\theta^x_{ji}(\bar{a}^x_{ji})^{\frac{1-\sigma_i}{\sigma_i}}\right]^{\sigma_i} (p_j)^{1-\sigma_i} + \left[\theta^v_i(\bar{a}^v_i)^{\frac{1-\sigma_i}{\sigma_i}}\right]^{\sigma_i} (p^{va}_i)^{1-\sigma_i}\right]^{\frac{1}{1-\sigma_i}} \\
&= \left[\sum_j \theta^x_{ji}\left(\frac{\bar{a}^x_{ji}}{\theta^x_{ji}}\right)^{1-\sigma_i} (p_j)^{1-\sigma_i} + \theta^v_i\left(\frac{\bar{a}^v_i}{\theta^v_i}\right)^{1-\sigma_i} (p^{va}_i)^{1-\sigma_i}\right]^{\frac{1}{1-\sigma_i}}
\end{aligned}
$$

ここで、$\bar{c}_i = \sum_j \bar{p}_j\bar{a}^x_{ji} + \bar{p}^{va}_i\bar{a}^v_i$ と $\theta^x_{ji} = \bar{p}_j\bar{a}^x_{ji}/\bar{c}_i$ の二つの関係を利用すると、次のように書き換えることができる。

$$
c_i(\mathbf{p}, p^{va}_i) = \left[\sum_j \theta^x_{ji}\left(\frac{\bar{c}_i}{\bar{p}_j}\right)^{1-\sigma_i} (p_j)^{1-\sigma_i} + \theta^v_i\left(\frac{\bar{c}_i}{\bar{p}^{va}_i}\right)^{1-\sigma_i} (p^{va}_i)^{1-\sigma_i}\right]^{\frac{1}{1-\sigma_i}}
$$

さらに書き換えると、結局次式を得る。

$$
c_i(\mathbf{p}, p^{va}_i) = \bar{c}_i\left[\sum_j \theta^x_{ji}\left(\frac{p_j}{\bar{p}_j}\right)^{1-\sigma_i} + \theta^v_i\left(\frac{p^{va}_i}{\bar{p}^{va}_i}\right)^{1-\sigma_i}\right]^{\frac{1}{1-\sigma_i}}
$$

この式が部門 i の CSF の単位費用関数である。やはり右辺にはベンチマークデータと代替の代替の弾力性しか含まれていない。

同様に、合成生産要素生産の単位費用関数は次式となる。

$$
c^{va}_i(\tilde{\mathbf{p}}^F_i) = \bar{c}^{va}_i\left[\sum_f \theta^F_{fi}\left[\frac{\tilde{p}^F_{fi}}{\bar{\tilde{p}}^F_{fi}}\right]^{1-\sigma^v_i}\right]^{\frac{1}{1-\sigma^v_i}}
$$

ただし、

$$
\begin{aligned}
\tilde{\mathbf{p}}^F_i &= \{\tilde{p}^F_{1i}, \cdots, \tilde{p}^F_{mi}\} = \{(1+t^F_{1i})p^F_1, \cdots, (1+t^F_{mi})p^F_m\} \\
\bar{c}^{va}_i &= \sum_f \bar{\tilde{p}}^F_{fi}\bar{a}^F_{fi} \\
\theta^F_{fi} &= \frac{\bar{\tilde{p}}^F_{fi}\bar{a}^F_{fi}}{\bar{c}^{va}_i}
\end{aligned}
$$

である。

支出関数 $e(\tilde{\mathbf{p}}^c, u)$（ただし、$\tilde{\mathbf{p}}^c = \{\tilde{p}^c_1, \cdots, \tilde{p}^c_n\} = \{(1+t^c_1)p_1, \cdots, (1+t^c_n)p_n\}$）についても、パラメータ γ_i にカリブレートした値 (23) を代入すればよい。

$$
\begin{aligned}
e(\tilde{\mathbf{p}}^c, u) &= u\left[\sum_i (\gamma_i)^{\sigma^c}(\tilde{p}^c_i)^{1-\sigma^c}\right]^{\frac{1}{1-\sigma^c}} \\
&= u\left[\sum_i \left[\theta^c_i\left(\frac{\bar{d}_i}{\bar{u}}\right)^{\frac{1-\sigma^c}{\sigma^c}}\right]^{\sigma^c} (\tilde{p}^c_i)^{1-\sigma^c}\right]^{\frac{1}{1-\sigma^c}} \\
&= u\left[\sum_i \theta^c_i\left(\frac{\bar{d}_i}{\bar{u}\theta^c_i}\right)^{1-\sigma^c} (\tilde{p}^c_i)^{1-\sigma^c}\right]^{\frac{1}{1-\sigma^c}}
\end{aligned}
$$

ここで $\bar{e} = \sum_i \bar{\tilde{p}}_i^c \bar{d}_i$、$\theta_i^c = \bar{\tilde{p}}_i^c \bar{d}_i / \bar{e}$ を使うと

$$e(\tilde{\mathbf{p}}^c, u) = u\frac{\bar{e}}{\bar{u}} \left[\sum_i \theta_i^c \left[\frac{\tilde{p}_i^c}{\bar{\tilde{p}}_i^c} \right]^{1-\sigma^c} \right]^{\frac{1}{1-\sigma^c}}$$

となる。これが CSF の支出関数である。

5.1.3 CSF の単位需要関数

同様に、CSF の単位需要関数を考える。元々の単位需要関数が次のように表現できた。

$$a_{ji}^x = \left[\frac{\alpha_{ji}^x}{p_j} \right]^{\sigma_i} \left[\sum_l (\alpha_{li}^x)^{\sigma_i} (p_l)^{1-\sigma_i} + (\alpha_i^v)^{\sigma_i} (p_i^{va})^{1-\sigma_i} \right]^{\frac{\sigma_i}{1-\sigma_i}} = \left[\frac{\alpha_{ji}^x c_i}{p_j} \right]^{\sigma_i}$$

$$a_i^v = \left[\frac{\alpha_i^v}{p_i^{va}} \right]^{\sigma_i} \left[\sum_l (\alpha_{li}^x)^{\sigma_i} (p_l)^{1-\sigma_i} + (\alpha_i^v)^{\sigma_i} (p_i^{va})^{1-\sigma_i} \right]^{\frac{\sigma_i}{1-\sigma_i}} = \left[\frac{\alpha_i^v c_i}{p_i^{va}} \right]^{\sigma_i}$$

再び、α_{ji}^x と α_i^v にカリブレートした値を代入する。まず、a_{ji}^x を考える。

$$\begin{aligned} a_{ji}^x &= \left[\frac{\theta_{ji}^x (\bar{a}_{ji}^x)^{\frac{1-\sigma_i}{\sigma_i}} c_i}{p_j} \right]^{\sigma_i} = \left[\frac{\bar{p}_j \bar{a}_{ji}^x}{\bar{c}_i} \frac{(\bar{a}_{ji}^x)^{\frac{1-\sigma_i}{\sigma_i}} c_i}{p_j} \right]^{\sigma_i} \\ &= \bar{a}_{ji}^x \left[\frac{\bar{p}_j}{\bar{c}_i} \frac{c_i}{p_j} \right]^{\sigma_i} = \bar{a}_{ji}^x \left[\frac{c_i / \bar{c}_i}{p_j / \bar{p}_j} \right]^{\sigma_i} \end{aligned}$$

同様に、α_i^v の値を元の式に代入する。

$$\begin{aligned} a_i^v &= \left[\frac{\theta_i^v (\bar{a}_i^v)^{\frac{1-\sigma_i}{\sigma_i}} c_i}{p_i^{va}} \right]^{\sigma_i} = \left[\frac{\bar{p}_i^{va} \bar{a}_i^v}{\bar{c}_i} \frac{(\bar{a}_i^{va})^{\frac{1-\sigma_i}{\sigma_i}} c_i}{p_i^{va}} \right]^{\sigma_i} \\ &= \bar{a}_i^v \left[\frac{\bar{p}_i^{va}}{\bar{c}_i} \frac{c_i}{p_i^{va}} \right]^{\sigma_i} = \bar{a}_i^v \left[\frac{c_i / \bar{c}_i}{p_i^{va} / \bar{p}_i^{va}} \right]^{\sigma_i} \end{aligned}$$

以上より、単位需要関数は次のように表現できる。

$$a_{ji}^x = \bar{a}_{ji}^x \left[\frac{c_i / \bar{c}_i}{p_j / \bar{p}_j} \right]^{\sigma_i} \qquad a_i^v = \bar{a}_i^v \left[\frac{c_i / \bar{c}_i}{p_i^{va} / \bar{p}_i^{va}} \right]^{\sigma_i}$$

これが CSF の単位中間投入需要関数と単位合成生産要素需要関数である。

同様にすると、単位生産要素需要関数の CSF を求めることができる。

$$a_{fi}^F = \bar{a}_{fi}^F \left[\frac{c_i^{va} / \bar{c}_i^{va}}{\tilde{p}_{fi}^F / \bar{\tilde{p}}_{fi}^F} \right]^{\sigma_i^v}$$

また、消費需要については次式となる。

$$d_i = u\frac{\bar{d}_i}{\bar{u}} \left[\frac{(e/u)/(\bar{e}/\bar{u})}{\tilde{p}_i^c / \bar{\tilde{p}}_i^c} \right]^{\sigma^c} = \bar{d}_i \left[\frac{u}{\bar{u}} \right]^{1-\sigma^c} \left[\frac{e/\bar{e}}{\tilde{p}_i^c / \bar{\tilde{p}}_i^c} \right]^{\sigma^c}$$

5.2 CSF（まとめ）

CSF の生産関数、効用関数

$$
\begin{aligned}
y_i &= \bar{y}_i \left[\sum_j \theta_{ji}^x \left(\frac{x_{ji}}{\bar{x}_{ji}} \right)^{\frac{\sigma_i-1}{\sigma_i}} + \theta_i^v \left(\frac{v_i^a}{\bar{v}_i^a} \right)^{\frac{\sigma_i-1}{\sigma_i}} \right]^{\frac{\sigma_i}{\sigma_i-1}} \\
v_i^a &= \bar{v}_i^a \left[\sum_f \theta_{fi}^F \left(\frac{v_{fi}}{\bar{v}_{fi}} \right)^{\frac{\sigma_i^v-1}{\sigma_i^v}} \right]^{\frac{\sigma_i^v}{\sigma_i^v-1}} \\
u &= \bar{u} \left[\sum_i \theta_i^C \left(\frac{d_i}{\bar{d}_i} \right)^{\frac{\sigma^c-1}{\sigma^c}} \right]^{\frac{\sigma^c}{\sigma^c-1}}
\end{aligned}
$$

CSF の単位費用関数、支出関数

$$
c_i(\mathbf{p}, p_i^{va}) = \bar{c}_i \left[\sum_j \theta_{ji}^x \left(\frac{p_j}{\bar{p}_j} \right)^{1-\sigma_i} + \theta_i^v \left(\frac{p_i^{va}}{\bar{p}_i^{va}} \right)^{1-\sigma_i} \right]^{\frac{1}{1-\sigma_i}} \tag{24}
$$

$$
c_i^{va}(\tilde{\mathbf{p}}_i^F) = \bar{c}_i^{va} \left[\sum_f \theta_{fi}^F \left[\frac{\tilde{p}_{fi}^F}{\bar{\tilde{p}}_{fi}^F} \right]^{1-\sigma_i^v} \right]^{\frac{1}{1-\sigma_i^v}} \tag{25}
$$

$$
e(\tilde{\mathbf{p}}^c, u) = u \frac{\bar{e}}{\bar{u}} \left[\sum_i \theta_i^c \left[\frac{\tilde{p}_i^c}{\bar{\tilde{p}}_i^c} \right]^{1-\sigma^c} \right]^{\frac{1}{1-\sigma^c}} \tag{26}
$$

CSF の単位需要関数、補償需要関数

$$
a_{ji}^x = \bar{a}_{ji}^x \left[\frac{c_i/\bar{c}_i}{p_j/\bar{p}_j} \right]^{\sigma_i} \tag{27}
$$

$$
a_i^v = \bar{a}_i^v \left[\frac{c_i/\bar{c}_i}{p_i^{va}/\bar{p}_i^{va}} \right]^{\sigma_i} \tag{28}
$$

$$
a_{fi}^F = \bar{a}_{fi}^F \left[\frac{c_i^{va}/\bar{c}_i^{va}}{\tilde{p}_{fi}^F/\bar{\tilde{p}}_{fi}^F} \right]^{\sigma_i^v} \tag{29}
$$

$$
d_i = \bar{d}_i \left[\frac{u}{\bar{u}} \right]^{1-\sigma^c} \left[\frac{e/\bar{e}}{\tilde{p}_i^c/\bar{\tilde{p}}_i^c} \right]^{\sigma^c} \tag{30}
$$

5.3 CSF の意義

CES 関数のパラメータをベンチマークデータによりカリブレートし、シミュレーションに利用する場合に、CSF という表現方法が利用できるということを説明してきたが、それではそもそも CSF を理解、あるいは利用する意義は何かを説明しておこう。

まず一つ目の利点として、既に指摘したことだが、シミュレーションのプログラムを簡潔にしやすくなるということがある。通常の表現方法では、ウェイトパラメータをカリブレートするための

プログラムを別途に記述しなければいけないが、CSF であればそのような必要はなくなり、プログラムを短かくしやすい。特に、モデルが大規模で多くの関数が利用されている場合には、これは大きな利点になる。

第二の利点は、CGE 分析についての重要な研究において CSF が利用されている場合があり、そのような研究をそのまま参考にできるという点である。具体的には、CGE 分析の研究者として著名な Thomas F. Rutherford や Christoph Böhringer 等はモデルの記述をおこなうさいに、CSF を利用することが多く、CSF を理解していれば彼等のプログラムや論文をそのまま参考にすることができる。また筆者（武田）も CSF を利用することが多く、CSF で記述したモデルのプログラムを多数公開している。

第三の利点は、MPSGE (mathematical programming system for general equilibrium) の記法を理解しやすくなるという点である。MPSGE とは Thomas F. Rutherford によって開発された GAMS のソルバーの一つで、数式を用いずに一般均衡モデルを GAMS 上で記述するための機能を提供してくれるものである[12)]。この MPSGE は内部では CSF を用いてモデルを記述しているため、CSF を理解しておくことで MPSGE の仕組みも理解しやすくなる。MPSGE については多くの研究で利用されているものであるので、そのプログラムを理解、利用しやくすなることは研究上の大きなメリットの一つとなる。

5.4 CSF の例

calibrated_share_form.gms は CSF でモデルを記述したプログラムである。モデル自体は calibration_example.gms と全く同じであり、プログラムの共通点も多いので、以下では相異点のみを説明する。

```
parameter
    cost_y0(i)          生産費用
    cost_v_a0(i)        合成生産要素生産の費用
    cost_c0             消費額
;

cost_y0(i) = sum(j, p0(j)*x0(j,i)) + p_va0(i)*v_a0(i);
cost_v_a0(i) = sum(f, p_f0(f)*vf0(f,i));
cost_c0 = sum(i, p0(i)*d0(i));

display cost_y0, cost_v_a0, cost_c0;
```

ここでは生産費用、消費額を表すパラメータを宣言している。後にシェアを求める際に利用する。

12) 通常の GAMS のソルバーは問題を解くためのアルゴリズムを実装したものであるが、MPSGE は単に特殊な記述方法を可能にするための機能である。MPSGE で記述したモデルは MCP タイプのモデルになるので、ソルバーとしては PATH 等の MCP ソルバーで解く。MPSGE もソルバーの一つと分類されているが、通常のソルバーとは役割が違う。MPSGE について詳しくは筆者による『MPSGE の利用方法』を見て欲しい。

シェアの計算においては本来 agent 価格、つまり企業や消費者が直面する価格を用いなければならない。このためここまでの説明では、価格として税込みの価格を考えてきたのであるが、ベンチマークデータでは全ての税率が 0 と仮定されている。従って、ベンチマークデータには税は出てこない。

```
parameter
    sh_x(j,i)    中間投入財のシェア
    sh_v(i)      合成生産要素のシェア
    sh_f(f,i)    生産要素のシェア
    sh_c(i)      消費における各財のシェア
;

sh_x(j,i) = p0(j)*x0(j,i) / cost_y0(i);
sh_v(i) = p_va0(i)*v_a0(i) / cost_y0(i);
sh_f(f,i) = p_f0(f)*vf0(f,i) / cost_v_a0(i);
sh_c(i) = p0(i)*d0(i) / cost_c0;

display sh_x, sh_v, sh_f, sh_c;
```

ここでシェアを表すパラメータを宣言し、値を代入している。

```
sh_x(j,i) = p0(j)*x0(j,i) / cost_y0(i);
```

これは生産費用に占める中間財 j の支出シェアである。

```
sh_v(i) = p_va0(i)*v_a0(i) / cost_y0(i);
```

これは生産費用に占める合成生産要素の支出シェアである。

```
sh_f(f,i) = p_f0(f)*vf0(f,i) / cost_v_a0(i);
```

これは合成生産要素生産における生産要素 f の支出シェアである。

```
sh_c(i) = p0(i)*d0(i) / cost_c0;
```

最後に、これは消費に占める財 i への支出シェアである。

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

*       生産の単位費用
e_c(i) .. c(i) =e=
          c0(i) * (sum(j, sh_x(j,i) * (p(j)/p0(j))**(1-sig(i)))
              + sh_v(i) *
          (p_va(i)/p_va0(i))**(1-sig(i)))**(1/(1-sig(i)));

*       合成生産要素生産の単位費用
e_c_va(i) ..
    c_va(i) =e=
    c_va0(i)
    * (sum(f,
        sh_f(f,i)
        * (((1+t_f(f,i))*p_f(f))/((1+t_f0(f,i))*p_f0(f)))**(1-sig_v(i)))
    )**(1/(1-sig_v(i)));

*       単位投入需要
e_a_x(j,i) ..
          a_x(j,i) =e= a_x0(j,i)
          * ((c(i) / c0(i)) / (p(j) / p0(j)))**sig(i);

*       単位合成生産要素需要
e_a_v(i) .. a_v(i) =e= a_v0(i)
            * ((c(i) / c0(i)) / (p_va(i) / p_va0(i)))**sig(i);

*       単位生産要素需要
e_a_f(f,i) ..
          a_f(f,i) =e= a_f0(f,i)
          * ((c_va(i) / c_va0(i)) /
              (((1+t_f(f,i))*p_f(f)) / ((1+t_f0(f,i))*p_f0(f))))**sig_v(i);

*       支出関数
e_e .. e =e=
       u * (e0/u0)
       * (sum(i, sh_c(i) * (((1+t_c(i))*p(i)) / ((1+t_c0(i))*p0(i)))**(1-sig_c))
       )**(1/(1-sig_c));

*       消費需要
e_d(i) .. d(i) =e=
```

```
            d0(i) * (u/u0)**(1-sig_c)
            * ((e/e0) / (((1+t_c(i))*p(i)) / ((1+t_c0(i))*p0(i))))**sig_c;
```

ここでモデルの式を定義している。式は calibration_example.gms とは異なるものだけを書いている。単位費用関数、支出関数、需要関数が CSF に変更されている。式については第 5.2 節の (24)〜(30) 式と比較して欲しい。

後は calibration_example.gms とほぼ同じである。実行してみて、calibration_example.gms と同じ計算結果になるか確認をして欲しい。普通の CES 関数の記述法と CSF による記述法という違いはあるが、同じデータを使って同じモデルの同じパラメータをカリブレートしているので、全く同じ計算結果になるはずである。

5.5 Cobb-Douglas 関数のケース

前節では CES 関数の CSF を見たが、同じことを Cobb-Douglas 関数にも適用することができる。以下、Cobb-Douglas 関数のケースの CSF を求める。

まず、部門 i の財の生産関数と部門 i の合成生産要素の生産関数を以下の Cobb-Douglas 関数とする。

$$
\begin{aligned}
y_i &= f_i^y(\mathbf{x}_i, v_i^a) = \phi_i \prod_i (x_{ji})^{\alpha_{ji}^x} (v_i^a)^{\alpha_i^v} \\
v_i^a &= f_i^v(\mathbf{v}_i) = \phi_i^v \prod_f (v_{fi})^{\beta_{fi}^v}
\end{aligned}
$$

ただし、$\sum_j \alpha_{ji}^x + \alpha_i^v = 1$、$\sum_f \beta_{fi}^v = 1$ である。また、効用関数を次の Cobb-Douglas 関数とする。

$$
u = \phi^u \prod_i (d_i)^{\gamma_i}
$$

ただし、$\sum_i \gamma_i = 1$ である。

この場合、単位費用関数と支出関数は次のような表現となる（第 A1 章を参照）

$$
c_i(\mathbf{p}_i, p_i^{va}) = \frac{1}{\phi_i} \prod_j \left[\frac{p_j}{\alpha_{ji}}\right]^{\alpha_{ji}} \left[\frac{p_i^{va}}{\alpha_i^v}\right]^{\alpha_i^v} \tag{31}
$$

$$
c_i^{va}(\tilde{\mathbf{p}}_i^F) = \frac{1}{\phi_i^v} \prod_f \left[\frac{\tilde{p}_{fi}^F}{\beta_{fi}^v}\right]^{\beta_{fi}^v} \tag{32}
$$

$$
e(\tilde{\mathbf{p}}^c, u) = \frac{u}{\phi^u} \prod_i \left[\frac{\tilde{p}_i^c}{\gamma_i^c}\right]^{\gamma_i^c} \tag{33}
$$

さらに、生産側の単位需要関数、消費の補償需要関数は次式となる。

$$
a_{ji}^x = \frac{\partial c_i(\mathbf{p}_i, p_i^{va})}{\partial p_j} = \left[\frac{\alpha_{ji}^x}{p_j}\right] \frac{1}{\phi_i} \prod_l \left[\frac{p_l}{\alpha_{li}^x}\right]^{\alpha_{li}^x} \left[\frac{p_i^{va}}{\alpha_{ji}^v}\right]^{\alpha_i^v} = \frac{\alpha_{ji}^x c_i}{p_j} \tag{34}
$$

$$
a_i^v = \frac{\partial c_i(\mathbf{p}_i, p_i^{va})}{\partial p_i^{va}} = \left[\frac{\alpha_i^v}{p_i^{va}}\right] \frac{1}{\phi_i} \prod_l \left[\frac{p_l}{\alpha_{li}^x}\right]^{\alpha_{li}^x} \left[\frac{p_i^{va}}{\alpha_{ji}^v}\right]^{\alpha_i^v} = \frac{\alpha_i^v c_i}{p_i^{va}} \tag{35}
$$

$$a_{fi}^F = \frac{\partial c_i^{va}(\tilde{\mathbf{p}}_i^F)}{\partial \tilde{p}_{fi}^F} = \left[\frac{\beta_{fi}^v}{\tilde{p}_{fi}^F}\right] \frac{1}{\phi_i^v} \prod_l \left[\frac{\tilde{p}_{li}^F}{\beta_{li}^v}\right]^{\beta_{li}^F} = \frac{\beta_{fi}^v c_i^{va}}{\tilde{p}_{fi}^F} \tag{36}$$

$$d_i = \frac{\partial e(\tilde{\mathbf{p}}^c, u)}{\partial \tilde{p}_i^c} = \left[\frac{\gamma_i^c}{\tilde{p}_i^c}\right] \frac{u}{\phi^u} \prod_j \left[\frac{\tilde{p}_j^c}{\gamma_j}\right]^{\gamma_j} = \frac{\gamma_i^c e}{\tilde{p}_i^c} \tag{37}$$

5.5.1 カリブレーション

Cobb-Douglas 関数の場合には、カリブレートするパラメータは以下の 2 つである。

1) ウェイトパラメータ（α_{ji}^x 等）
2) スケールパラメータ（ϕ_i 等）

ウェイトパラメータ

ウェイトパラメータは (34)〜 (37) 式の（単位）需要関数を用いるとカリブレートできる。例えば、α_{ji}^x は、(34) 式より

$$\alpha_{ji}^x = \frac{\bar{p}_j \bar{a}_{ji}^x}{\bar{c}_i}$$

により計算できる。Cobb-Douglas 関数の場合、ウェイトパラメータ α_{ji}^x はベンチマークの費用における中間財 j への支出シェアを表すことがわかる。同様に、他のウェイトパラメータもカリブレートできる。

$$\alpha_i^v = \frac{\bar{p}_i^{va} \bar{a}_i^v}{\bar{c}_i} \qquad \beta_{fi}^v = \frac{\bar{\tilde{p}}_{fi}^F \bar{a}_{fi}^F}{\bar{c}_i^{va}} \qquad \gamma_i = \frac{\bar{\tilde{p}}_i^c \bar{d}_i}{\bar{e}}$$

スケールパラメータ

ウェイトパラメータが決まればスケールパラメータは生産関数、効用関数から次のようにカリブレートできる。

$$\phi_i = \bar{y}_i \Big/ \prod_i (\bar{x}_{ji})^{\alpha_{ji}^x} (\bar{v}_i^a)^{\alpha_i^v}$$

$$\phi_i^v = \bar{v}_i^a \Big/ \prod_f (\bar{v}_{fi})^{\beta_{fi}^v}$$

$$\phi^u = \bar{u} \Big/ \prod_i (\bar{d}_i)^{\gamma_i}$$

あるいは、(31)〜(33) 式を使ってカリブレートしても同じである。

$$\phi_i = \frac{1}{\bar{c}_i} \prod_j \left[\frac{\bar{p}_j}{\alpha_{ji}}\right]^{\alpha_{ji}} \left[\frac{\bar{p}_i^{va}}{\alpha_{ji}^v}\right]^{\alpha_i^v}$$

$$\phi_i^v = \frac{1}{\bar{c}_i^{va}} \prod_f \left[\frac{\bar{\tilde{p}}_{fi}^F}{\beta_{fi}^v}\right]^{\beta_{fi}^v}$$

$$\phi^u = \frac{\bar{u}}{\bar{e}} \prod_i \left[\frac{\bar{\tilde{p}}_i^c}{\gamma_i^c}\right]^{\gamma_i^c}$$

5.5.2 CSF

CES 関数のときと同様に、カリブレートされた値を元の関数に入れてやれば、calibarated share form を求めることができる。

CSF の生産関数、効用関数

$$y_i = \bar{y}_i \prod_j \left[\frac{x_{ji}}{\bar{x}_{ji}}\right]^{\alpha^x_{ji}} \left[\frac{v^a_i}{\bar{v}_i}\right]^{\alpha^v_i}$$
$$v^a_i = \bar{v}^a_i \prod_f \left[\frac{v_{fi}}{\bar{v}_{fi}}\right]^{\beta^v_{fi}}$$
$$u = \bar{u} \prod_i \left[\frac{d_i}{\bar{d}_i}\right]^{\gamma_i}$$

CSF の単位費用関数、支出関数

$$c_i(\mathbf{p}, p^{va}_i) = \bar{c}_i \prod_j \left[\frac{p_j}{\bar{p}_j}\right]^{\alpha^x_{ji}} \left[\frac{p^{va}_i}{\bar{p}^{va}_i}\right]^{\alpha^{va}_i}$$
$$c^{va}_i(\tilde{\mathbf{p}}^F_i) = \bar{c}^{va}_i \prod_f \left[\frac{\tilde{p}^F_{fi}}{\bar{\tilde{p}}^F_{fi}}\right]^{\beta^v_{fi}}$$
$$e(\tilde{\mathbf{p}}^c, u) = u\frac{\bar{e}}{\bar{u}} \prod_i \left[\frac{\tilde{p}_i}{\bar{\tilde{p}}_i}\right]^{\gamma_i}$$

CSF の単位需要関数、補償需要関数については、CES 関数のケースである (27)〜(30) 式において代替の弾力性を 1 と置いたものに等しくなる。

5.5.3 プログラム例

`calibrated_share_form_CD.gms` と `calibration_example_CD.gms` が生産関数、効用関数が Cobb-Douglas 関数である場合のプログラム例である。基本的に、`calibrated_share_form.gms` と `calibration_example.gms` と同じで、関数形だけが異なっている。

6 終わりに

この章では、CGE 分析における関数形の選択と関数のパラメータの特定化について説明した。関数形には選択肢がたくさんあるが、基本的には CES 関数が用いられることが多い。CES 関数を利用する際には、そのウェイトパラメータを特定化する必要があるが、これには通常「カリブレーション」という手法が用いられる。この章の手法は、第 9 章以後で用いていく。

参考文献

Ballard, Charles L., John B. Shoven, and John Whalley (1985) "General Equilibrium Computations of the Marginal Welfare Costs of Taxes in the United States," *American Economic Review*, Vol. 75, No. 1, pp. 128–138.

Chateau, Jean, Rob Dellink, and Elisa Lanzi (2014) "An Overview of the OECD ENV-Linkages Model: Version 3," *OECD Environment Working Papers, No. 65*, No. 65, p. 43, DOI: 10.1787/19970900.

Chen, Y-H Henry, Sergey Paltsev, John Reilly, Jennifer Morris, and Mustafa H. Babiker (2015) "The MIT EPPA6 Model: Economic Growth, Energy Use, Emissions, and Food Consumptions," No. 278, p. 40.

Dixon, Peter B. and Dale W. Jorgenson (2013) "Introduction," in Dixon, Peter B and Dale W Jorgenson eds. *Handbook of Computable General Equilibrium Modeling*, Vol. 1, pp. 1–22: Elsevier, DOI: 10.1016/B978-0-444-59568-3.00001-8.

Hertel, Thomas W. (1999) *Global Trade Analysis: Modeling and Applications*, New York: Cambridge University Press, URL: `http://econpapers.repec.org/RePEc:cup:cbooks:9780521643740`.

Hertel, Thomas W., J. Mark Horridge, and K.R. Pearson (1992) "Mending the Family Tree a Reconciliation of the Linearization and Levels Schools of AGE Modelling," *Economic Modelling*, Vol. 9, No. 4, pp. 385–407, October, DOI: 10.1016/0264-9993(92)90020-3.

Johansen, Leif (1960) *A Multi-Sectoral Study of Economic Growth*, Amsterdam: North-Holland Publishing Co.

Lofgren, Hans, Rebecca Lee Harris, and Sherman Robinson (2002) "A Standard Computable General Equilibrium (CGE) Model in GAMS,"Technical report, International Food Policy Research Institute (IFPRI), Washington, D.C., URL: `https://www.ifpri.org/publication/standard-computable-general-equilibrium-cge-model-gams`.

Robichaud, Véronique, André Lemelin, Hélèn Maisonnave, and Vernard Decaluwé (2012) "No PEP-1-1 A User Guide,"Technical report, URL: `http://www.pep-net.org/programs/mpia/pep-standard-cge-models/`.

Shoven, John B. and John Whalley (1992) *Applying General Equilibrium*, New York: Cambridge University Press.

van der Mensbrugghe, Dominique (2005) "LINKAGE Technical Reference Document Version 6.0,"Technical report, October, 2001 January, 2005.

市岡修 (1991) 『応用一般均衡分析』, 有斐閣.

7 補足

7.1 多段階の CES 関数における代替の弾力性の値

第 2.2 節で多段階の CES 関数を紹介した。多段階の CES 関数を利用する場合、各段階の代替の弾力性の値には異なる値を指定する必要がある。というのは、各段階の代替の弾力性の値に同じ値を設定すると結局 1 段階の CES 関数を利用しているのと同じになってしまうからである。

例えば、(1)、(2) 式の二段階の CES 関数を考える。ここで、1 段階目の代替の弾力性と 2 段階目の代替の弾力性が等しい、つまり $\sigma_i^x = \sigma$ であるとする。すると (1) は次のように書き換えられる。

$$y = \left[\sum_i \beta_i \left(\left[\sum_j \delta_{ji} (z_{ji})^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}} \right)^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}} = \left[\sum_{i,j} \beta_i \delta_{ji} (z_{ji})^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}}$$

ここで $\zeta_{ji} = \beta_i \delta ji$ と置けば、

$$y = \left[\sum_{i,j} \zeta_{ji} (z_{ji})^{\frac{\sigma-1}{\sigma}} \right]^{\frac{\sigma}{\sigma-1}}$$

となり、代替の弾力性 σ の 1 段階の CES 関数となる。

多段階の CES 関数を利用するのなら各段階における代替の弾力性の値に異なる値を想定しないと無意味ということである。

7.2 Harberger Convention のチェック

第 4.3.1 節において、“Harberger Convention” について説明した。そして、そこで、**「分析において変数の変化率のみを見る場合には、ベンチマークの価格をどのような値に置いても分析結果は変わらない」**と述べた。これが実際に成り立つかどうかをチェックしてみよう。

以下、`harberger_check.gms` と `harberger_check_2.gms` を利用する。`harberger_check.gms` は `calibration_example.gms` とほぼ同じである。ただし、結果を表示する部分のみ少し変更している。

`harberger_check.gms` では、何もショックを与えない初期均衡（プログラムでは `bench`）の計算に加え、

- 資本の賦存量を 20% 削減するシミュレーション（プログラムでは `cap_up`）
- `man` と `ser` という 2 つの財に対して、10% の消費税を導入するシミュレーション（`tax_c`）

という二つのシミュレーションをおこなっている。

`harberger_check.gms` では次のマクロを定義している。

```
$macro calc_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("v_y_agr",sc) = p.l("agr")*y.l("agr") / p.l("agr"); \
    results("v_y_man",sc) = p.l("man")*y.l("man") / p.l("agr"); \
    results("v_y_ser",sc) = p.l("ser")*y.l("ser") / p.l("agr"); \
    results("d_agr",sc) = d.l("agr"); \
    results("d_man",sc) = d.l("man"); \
    results("d_ser",sc) = d.l("ser");
```

results("u",sc) には効用の水準、results("y_agr",sc) からの 3 つには各部門の生産量を代入している。results("v_y_agr",sc) には AGR で測った AGR の実質（相対）生産額を代入している。その次の二つも同様である。最後の 3 つには各財の需要量を代入している。

以下が、harberger_check.gms におけるパラメータ results の値である。

```
----    413 PARAMETER results  結果を表示するためのパラメータ

              bench      cap_up       tax_c

u           360.000     320.000     359.873
y_agr       140.000     127.327     143.169
y_man       300.000     263.079     297.771
y_ser       150.000     136.085     149.366
v_y_agr     140.000     127.327     143.169
v_y_man     300.000     282.970     297.399
v_y_ser     150.000     136.759     149.354
d_agr        70.000      63.663      72.698
d_man       220.000     192.925     217.983
d_ser        70.000      63.506      69.318
```

さらに、以下は各変数の bench の値からの変化率（%）である。

```
----    422 PARAMETER results_pc  変化率を表示するためのパラメータ

             cap_up       tax_c

u           -11.111      -0.035
y_agr        -9.052       2.263
y_man       -12.307      -0.743
y_ser        -9.277      -0.422
v_y_agr      -9.052       2.263
v_y_man      -5.677      -0.867
v_y_ser      -8.827      -0.431
d_agr        -9.052       3.854
d_man       -12.307      -0.917
d_ser        -9.277      -0.975
```

これに対し、harberger_check_2.gms では Harberger Convention を利用せずに、ベンチマーク均衡における価格を以下のように設定している。

```
*       Harberger convention を使わない。
p0(i) = 2;
p_va0(i) = 0.5;
p_f0(f) = 3;
```

この harberger_check_2.gms の計算結果が次の通りである。

```
----    418 PARAMETER results  結果を表示するためのパラメータ

             bench      cap_up       tax_c

u         1080.000     960.000    1079.620
y_agr      210.000     190.990     214.753
y_man      450.000     394.619     446.657
y_ser      225.000     204.128     224.050
v_y_agr    210.000     190.990     214.753
v_y_man    450.000     424.456     446.099
v_y_ser    225.000     205.139     224.031
d_agr      105.000      95.495     109.047
d_man      330.000     289.387     326.974
d_ser      105.000      95.260     103.977


----    428 PARAMETER results_pc  変化率を表示するためのパラメータ

            cap_up       tax_c

u          -11.111      -0.035
y_agr       -9.052       2.263
y_man      -12.307      -0.743
y_ser       -9.277      -0.422
v_y_agr     -9.052       2.263
v_y_man     -5.677      -0.867
v_y_ser     -8.827      -0.431
d_agr       -9.052       3.854
d_man      -12.307      -0.917
d_ser       -9.277      -0.975
```

results パラメータの値は、変数の値の水準を表しており、harberger_check.gms とは全く違う値になっている。これは当たり前の結果である。しかし、変数の変化率を表す results_pc パラ

メータの値は harberger_check.gms のときの全く同じ値になっていることがわかる。つまり、変数の変化率については、ベンチマーク均衡における価格をどのような水準に設定しても変わらないということである。これより、第 4.3.1 節で述べたように、**「あるショックに対して内生変数が何%変化するのかということを分析したいのなら、ベンチマーク均衡における価格をどのような水準に設定してもよい」**ということになる。ただし、実際には 1 と置くことが普通である。

8　履歴

- 2022-01-20: 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。
- 2016-05-14: 7.1 節を追加
- 2015-11-21: 5.3 節を追加。
- 2015-11-20: 説明修正
- 2015-10-23: 誤植の修正