産業動学に関する研究ノート（数値計算編）（3）

加　藤　浩

6．関数 profit

mc：限界費用 *mc*（ベルトラン競争）
M：市場規模 *M*（ベルトラン競争）
wstar：効用曲線の屈曲点 ω^*（ベルトラン競争）
ggamma：限界費用のパラメータ γ（クールノー競争）
D：市場規模 *D*（クールノー競争）
f：固定費用 *f*（クールノー競争）
beta：割引因子 β

表５　パラメータの一覧

descn：産業構造の総数
pstar：均衡価格（nfirms 次元ベクトル）
theta：限界費用（クールノー競争）（nfirms 次元ベクトル）
quan：均衡生産量（クールノー競争）（nfirms 次元ベクトル）
sigma：購入確率（ベルトラン競争）（nfirms 次元ベクトル）
profstar：均衡利潤（nfirms 次元ベクトル）
profit：利潤行列（descn×nfirms 行列）（a.(PREFIX)_pr(nfirms).mat に保存）
agprof：利潤行列（descn×nfirms 行列）（a.(PREFIX)_cons(nfirms).mat に保存）
share：市場シェア行列（descn×nfirms 行列）
pmcmarg：利潤マージン・ベクトル（descn 次元ベクトル）
concent：１社集中度ベクトル（descn 次元ベクトル）

表６　変数の一覧

```
function [] = profit()
  global rlnfirms kmax binom
  global mc M wstar w egw p profstar sigma D f ggamma quan
         theta pstar
  global profit agprof share pmcmarg concent

  nfirms = 1;
  while nfirms <= rlnfirms
    descn = binom(nfirms+kmax+1, kmax+2);

    profit = zeros(descn, nfirms);
    agprof = zeros(descn, nfirms);
    share = zeros(descn, nfirms);
    pmcmarg = zeros(descn, 1);
    concent = zeros(descn, 1);

    if 'ベルトラン競争'
      w = [];
      egw = [];
      pstar = [];
      profstar = [];
      sigma = zeros(nfirms, 1);
      p = 5.5*ones(nfirms, 1);

      cqprofit(nfirms, descn);

    elseif 'クールノー競争'
      quan = [];
      profstar = [];
      w = [];
```

```
        theta = [];
        pstar = [];

        ccprofit(nfirms, descn);
      end

      save(['a.' PREFIX '_pr' int2str(nfirms) '.mat'],
      'profit');
      save(['a.' PREFIX '_cons' int2str(nfirms) '.mat'], ...
        'agprof', 'share', 'pmcmarg', 'concent');

      nfirms = nfirms+1;
    end
  end
```

nfirms 社の企業の間で展開されるクールノー競争，もしくはベルトラン競争の均衡利潤について，企業数 nfirms を 1 社から rlnfirms 社まで変えてそれぞれ計算する[62]。エンコードされた産業構造を行番号とし，企業のインデックスを列番号とした利潤行列に，対応する均衡利潤を格納する。こうして構築された利潤行列をファイルに保存する[63]。このファイルはメインルーチンで読み込まれ，利潤行列は割引現在価値を計算する際に使用される[64]。またこの関数では，各企

62) クールノー競争で各企業が設定する生産量，あるいはベルトラン競争で各企業が設定する価格は，産業動学に影響を与えない（静学・動学分解（dynamic-static breakdown））。産業動学において産業構造は変化するが，実現した産業構造に対応して均衡生産量，あるいは均衡価格が定まる。つまり，生産量や価格の決定は産業構造を所与とした静学的な問題となり，投資決定や退出決定と切り離して考える。数値計算の観点に立つと，均衡利潤を計算するアルゴリズムは，マルコフ完全ナッシュ均衡値を計算するアルゴリズムから独立したオフライン・アルゴリズムになる。

63) 1 社問題に用いる利潤行列から rlnfirms 社問題の利潤行列まで，rlnfirms 個のファイルが作成される。

64) 具体的には，nfirms社問題のループにおいて，nfirms社の利潤行列が保存されているファイルを読み込む。読み込まれたデータは利潤行列profitに格納され，関数optimizeにおいて，最大化された期待割引現在価値nvalを計算するときに使用される（5.8節参照）。

業の市場シェア，各企業の利潤マージン，1社集中度といった統計量も計算してファイルに保存する。

メインルーチンで設定されたパラメータ rlnfirms と kmax，およびメインルーチンで構築された Binom 行列 binom をグローバル変数として，この関数で共有する(02行目)。nfirms＝1から始めて, nfirms＝rlnfirms までループし，各企業の均衡利潤を計算する。(54)式に従って，企業数が nfirms，最大の効率性水準が kmax のときに実現可能な産業構造の総数 descn を, Binom 行列から計算する（08行目）。さらに各統計量を初期化する（10～14行目）。

利潤行列を計算する処理は，事前に選択された競争形態に応じて，ベルトラン競争のケース（16～24行目），あるいは，クールノー競争のケース（26～33行目）のいずれかに分岐する[65]。ベルトラン競争を選択した場合は，以下の処理が行われる。最初に，均衡利潤の計算で用いる変数を初期化する（17～21行目）[66]。また，ニュートン＝ラフソン法の初期点を p = (0.5, …, 0.5)と設定する（22行目）。そうした上で，関数 cqprofit を呼び出して，nfirms 社の均衡利潤を descn 個の産業構造すべてについて計算する（24行目）。クールノー競争を選択した場合も同様の処理が行われる。まず，均衡利潤の計算で使用する変数を初期化する（27～31行目）。関数 ccprofit を呼び出して，nfirms 社の利潤行列を descn 個の産業構造すべてについて計算する（33行目）。こうして計算された利潤行列 profit をファイルに保存する（36行目）[67]。また，同時に計算された統計量 agprof, share, pmcmarg, concent を別のファイルに保存する（37, 38行目）[68]。nfirms を1つ増やして次のループに進む（40行目）。

---

65) PREFIX＝'c'ならばクールノー競争の利潤を，PREFIX＝'b'ならばベルトラン競争の利潤を計算するというように，PREFIX の値に応じて処理を分岐させる。

66) a = []は空配列を a に代入する処理である。

67) 例えば，3社のベルトラン競争のデータは，a.b_pr3.mat に保存される。

68) 例えば，3社のクールノー競争のデータは，a.c_cons3.mat に保存される。

6.1　クールノー競争

6.1.1　モデル

同質財を生産・販売する N 社の企業の間で，クールノー競争が展開される。効率性パラメータ ω_n は，企業 n の限界費用の大きさを決める。限界費用は生産量に対して一定であり，ω_n の実現値に影響を受ける。つまり，

$$MC = \theta(\omega_n), \quad \theta' < 0. \tag{146}$$

ω_n は次式に従って推移する。

$$\omega_n' = \omega_n + \tau - \nu. \tag{147}$$

ここで，$\nu \in \{0, 1\}$ は産業全体のショック，$\tau \in \{0, 1\}$ は企業固有のショックである[69)]。τ が上昇するとき，あるいは ν が低下するときに ω_n は増加し，これにより限界費用は低下する。

$\tau \in \{0, 1\}$ の各値が実現する確率は投資水準 x に影響を受ける。τ の確率分布を次のように特定化する。

$$\Pr(\tau) = \begin{cases} \dfrac{ax}{1+ax} & （\tau = 1 のとき） \\ \dfrac{1}{1+ax} & （\tau = 0 のとき）. \end{cases} \tag{148}$$

a は定数である。$\delta \in [0, 1]$ は定数として，ν の確率分布を次のように特定化する。

$$\Pr(\nu) = \begin{cases} \delta & （\nu = 1 のとき） \\ 1 - \delta & （\nu = 0 のとき）. \end{cases} \tag{149}$$

企業 n の生産量を q_n とすると，産業全体の総生産量 Q は次のようになる。

$$Q = \sum_{n=1}^{N} q_n. \tag{150}$$

市場需要関数を以下のように定める。

69) 例えば，ν は生産要素価格（対数値），τ は各企業の生産効率性（対数値）と解釈できる。

$$P(Q) = D - Q. \tag{151}$$

D は定数であり，その大きさは市場規模を表す。固定費用を f とすると，企業 n の利潤は次のようになる。

$$\pi_n = P(Q)q_n - \theta(\omega_n)q_n - f. \tag{152}$$

クールノー均衡における市場価格と各企業の生産量は，次式として導かれる。

$$P^* = \frac{1}{N^*+1}\left(D + \sum_{n=1}^{N^*}\theta(\omega_n)\right), \tag{153}$$

$$q^*(\omega_n) = \mathrm{Max}\{0, P^* - \theta(\omega_n)\}. \tag{154}$$

ここで，N^*は均衡において生産量が正となる企業の総数である。企業 n の均衡利潤は，

$$\pi^* = \mathrm{Max}\{-f, \{P^* - \theta(\omega_n)\}^2 - f\} \tag{155}$$

となる。均衡生産量がゼロとなるときは，固定費用に等しい損失が発生する。

数値計算のために，限界費用関数θを次のように特定化する。

$$\theta(\omega_n) = \gamma\exp\{-\omega_n\}. \tag{156}$$

γは定数である。

6.1.2　コード

```
function [] = ccprofit(nfirms, descn)
  global w ggamma D f quan profstar theta pstar
  global profit agprof share pmcmarg concent

  i = 1;
  while i <= descn
    w = qdecode(i);
```

```
    w = w-4;
    theta = ggamma*exp(-w);
    n = nfirms;
    p = (D + sum(theta(1:n)))/(n+1);

    while ~((p - theta(n) >= 0) || (n == 1))
      n = n-1;
      p = (D + sum(theta(1:n)))/(n+1);
    end

    q = zeros(nfirms, 1);

    if p - theta(n) > 0
      q(1:n) = p - theta(1:n);
    end

    quan = q;

    pstar = D - sum(quan);
    profstar = (pstar > theta).*(pstar-theta).*quan - f;
    profit(i, :) = profstar';
    share(i, :) = quan';

    if sum(quan) > 0
      pmcmarg(i) = pstar/(sum(theta.*quan))*sum(quan);
      concent(i) = max(quan)/sum(quan);
    else
      pmcmarg(i) = 1;
    end
```

```
37
38           i = i+1;
39         end
40       end
```

エンコードされた産業構造 i = 1, ⋯, descn の下, nfirms 社の企業の間でクールノー競争が繰り広げられる。各企業が獲得する均衡利潤を計算して，利潤行列の第 i 行に格納する。また，市場シェア，利潤マージン，集中度といった統計量も同時に計算する。

まず，産業構造 i をデコードして w を得る（07行目）。得られた産業構造ベクトル w に対して，w − 4と変換する（08行目）[70]。$\mathtt{w} = (\omega_1, \cdots, \omega_{\mathtt{nfirms}})$から限界費用 $\mathtt{theta} = (\gamma\exp\{-\omega_1\}, \cdots, \gamma\exp\{-\omega_{\mathtt{nfirms}}\})$を計算する（09行目）。変数 n に nfirms を代入した上で（10行目），n 社の企業が生産しているときに実現する市場価格 p を計算する（11行目）。ただし，各企業の生産量の中には，マイナスの水準も含まれている可能性がある。すなわち，市場価格 p よりも高い限界費用 theta を持つ企業の生産量はマイナスとなり，このような生産量も含めて，ひとまず市場価格 p を計算する。13～16行目のループで，生産量がマイナスとなる企業を除外して市場価格を再計算する。

$$\mathtt{theta}(1) \leq \mathtt{theta}(2) \leq \cdots \leq \mathtt{theta}(\mathtt{n}) \tag{157}$$

であるから，

$$\mathtt{q}(1) \geq \mathtt{q}(2) \geq \cdots \geq \mathtt{q}(\mathtt{n}) \tag{158}$$

となる。したがって，n 番目の企業から始めて，n − 1番目の企業，n − 2番目の企業と順次，生産量がマイナスになっているかどうかを検討していく。生産量がマイナスとなるときは，均衡生産量をゼロとするので，この企業を除外して市場価格を再度計算する。13行目の **while** ループの条件式は，

70) つまり，$\omega_n \in \{0, 1, 2, \cdots\}$を$\omega_n \in \{-4, -3, -2, \cdots\}$と変換して限界費用を計算する。

```
~((p - theta(n) >= 0) || (n == 1))
= (p - theta(n) < 0) && (n > 1)
```
(159)

である。つまり，企業数 n が２社以上であり，かつ n 番目の企業の限界費用 theta(n)が市場価格 p よりも高い限り，**while** ループ内の処理を繰り返す。このとき，n 番目の企業の生産量はマイナスになるので，この企業を除外し（14行目），企業数が n－1のときの市場価格 p を再計算する（15行目）。再び **while** ループの条件式に戻り，n－1番目の企業の限界費用が再計算された市場価格よりも高くなるかどうかを判定する。**while** ループから抜け出たときの n の値は，生産量が非負となる企業の総数である。

次に，各企業の均衡生産量 quan を求める。p - theta＞0となる企業の均衡生産量を計算する（21行目）。p - theta ≤ 0となる企業については，18行目の初期化が維持されるので，均衡生産量はゼロとなる。均衡総生産量 sum(quan) を市場需要関数に代入して，均衡価格 pstar を計算する（26行目）。続いて均衡利潤 profstar を計算する（27行目）。pstar ≤ theta となる企業の均衡生産量はゼロであるから，固定費用に等しい損失を profstar に代入する。こうして計算された産業構造 i の均衡利潤 profstar を利潤行列 profit の第 i 行に（28行目），均衡生産量 quan を市場シェア行列 share の第 i 行に（29行目），それぞれ格納する。

均衡総生産量が正であるとき，産業構造 i における利潤マージン

$$\text{pmcmarg} = \frac{\sum_{n=1}^{N*} P^* q^*(\omega_n)}{\sum_{n=1}^{N*} \theta(\omega_n) q^*(\omega_n)} \tag{160}$$

と，１社集中度（CR1）

$$\text{concent} = \frac{\operatorname*{Max}_{n} q^*(\omega_n)}{\sum_{n=1}^{N*} q^*(\omega_n)} \tag{161}$$

を計算して，利潤マージン・ベクトル pmcmarg の第 i 要素（32行目），および１

社集中度ベクトル concent の第 i 要素（33行目）に，それぞれ格納する。均衡総生産量がゼロであるときは，すべての企業が $P^* = \theta(\omega_n)$となっているので，利潤マージンを１とし（35行目），１社集中度ベクトルはゼロとする。

6.2　ベルトラン競争

6.2.1　モデル

差別化された財を生産・販売する N 社の企業の間で，ベルトラン競争が繰り広げられる。企業 n が生産する財を財 n と呼ぶ。また，M 人の消費者が財を購入する。各々の消費者は，差別化された N 種類の財 j（$j = 1, \cdots, N$）と外部財0の中から１つを選択して，１単位購入する71)。代表的な消費者が財 j（$j = 0, 1, \cdots, N$）を購入したときに得る効用を，次のように置く72)。

$$U_j = v_j - r_j + \varepsilon_j. \tag{162}$$

v_j は財 j の品質，r_j は財 j の価格であり，ε_j は各消費者の好みの違いを表わす確率変数である。確率変数ε_j は累積分布関数

$$F(x) = \Pr(\varepsilon_j \le x) = \exp\left\{-\exp\left\{-\left(\frac{x}{\mu}+\eta\right)\right\}\right\} \tag{163}$$

に従う73)。μ, ηは定数である。確率密度関数は，

$$f(x) = \frac{1}{\mu}\exp\left\{-\left(\frac{x}{\mu}+\eta\right)\right\}\exp\left\{-\exp\left\{-\left(\frac{x}{\mu}+\eta\right)\right\}\right\} \tag{164}$$

となる。$\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_N$ は，同一の分布から独立に抽出される。

財 j に対する需要量を求める。財 j を購入することが最適となる条件は，次のようになる。

---

71) 外部財とは，財 j（$j = 1, \cdots, N$）以外の他の全ての財の合成財である。財 j（$j = 1, \cdots, N$）のどれも選択しないときは，外部財を選択すると考える。

72) 消費者行動を離散選択（discrete choice）モデルで考える（Anderson&de Palma&Thisse (1992)，Train(2009)参照）。

73) ガンベル分布と呼ばれる。同一の分布に従う n 個の独立な確率変数の中で，最大値をとる確率変数の極限分布（$n \to \infty$のときの漸近分布）がガンベル分布となる。

$$U_j \geq U_q \ (q = 0, 1, \cdots, N). \tag{165}$$

（162）式を代入すると，

$$\begin{aligned}\varepsilon_j - \varepsilon_q &\geq v_q - v_j + r_j - r_q \\ &= (v_q - v_0) - (v_j - v_0) + (r_j - r_0) - (r_q - r_0) \ (q = 0, 1, \cdots, N).\end{aligned} \tag{166}$$

ここで，

$$g(\omega_i) \equiv v_i - v_0 \ (i = 0, 1, \cdots, N) \tag{167}$$

と定義する。$g(\omega_i)$は外部財0を基準とした財 i の相対品質である。g は有界な関数であり，$g' > 0$，$g'' < 0$を満たす。効率性パラメータω_iは相対品質を決定し，財 i の需要に影響を与える。ω_i は確率変数であり，（147）式に従って推移する。τは（148）式，νは（149）式で与えられる確率分布に従う[74]。

外部財0を基準とした財 i の相対価格 p_i を次のように定める。

$$p_i \equiv r_i - r_0 \ (i = 0, 1, \cdots, N). \tag{168}$$

これらの記号を用いると，（166）式は，

$$\varepsilon_q - \varepsilon_j \leq \{g(\omega_j) - p_j\} - \{g(\omega_q) - p_q\} \ (q = 0, 1, \cdots, N) \tag{169}$$

と書ける。（169）式を満たすような$\varepsilon_0, \cdots, \varepsilon_N$が実現する確率$\sigma_j$は，$\varepsilon_0, \cdots, \varepsilon_N$が独立した確率変数であることから，

$$\begin{aligned}\sigma_j = &\Pr(\varepsilon_0 - \varepsilon_j \leq \{g(\omega_j) - p_j\} - \{g(\omega_0) - p_0\}) \\ &\times \Pr(\varepsilon_1 - \varepsilon_j \leq \{g(\omega_j) - p_j\} - \{g(\omega_1) - p_1\}) \\ &\times \cdots \times \Pr(\varepsilon_{j-1} - \varepsilon_j \leq \{g(\omega_j) - p_j\} - \{g(\omega_{j-1}) - p_{j-1}\}) \\ &\times \Pr(\varepsilon_{j+1} - \varepsilon_j \leq \{g(\omega_j) - p_j\} - \{g(\omega_{j+1}) - p_{j+1}\}) \\ &\times \cdots \times \Pr(\varepsilon_N - \varepsilon_j \leq \{g(\omega_j) - p_j\} - \{g(\omega_N) - p_N\})\end{aligned} \tag{170}$$

---

74) $\tau = 1$ が実現するとき，ω_iは上昇するので，相対品質も上昇する。これは，財 i の品質 v_i が上昇したことと同義である。$\nu = 1$ が実現するとき，ω_iは低下するので，相対品質も低下する。これは，外部財の品質 v_0 が上昇したことと同義である。

となる。σ_j は代表的消費者が財 j を購入する確率なので，$M\sigma_j$ が財 j の需要量となる。$\varepsilon_0, \cdots, \varepsilon_N$ は共通の累積分布関数 F に従う確率変数であることから，$\varepsilon_j = x$ が実現したときに財 j を購入する確率は，

$$\begin{aligned}&\Pr(\varepsilon_0 \le \{g(\omega_j) - p_j\} - \{g(\omega_0) - p_0\} + x)\\&\times\cdots\times \Pr(\varepsilon_N \le \{g(\omega_j) - p_j\} - \{g(\omega_N) - p_N\} + x)\\&= \prod_{q \ne j} F(\{g(\omega_j) - p_j\} - \{g(\omega_q) - p_q\} + x)\end{aligned} \tag{171}$$

となる。$\varepsilon_j = x$ の確率密度は $f(x)$ であるから，σ_j は次のようになる。

$$\sigma_j = \int_{-\infty}^{\infty} f(x) \prod_{q \ne j} F(\{g(\omega_j) - p_j\} - \{g(\omega_q) - p_q\} + x) dx \cdot \tag{172}$$

（163）式より，

$$\begin{aligned}&F(\{g(\omega_j) - p_j\} - \{g(\omega_q) - p_q\} + x)\\&= \exp\left\{-\exp\left\{-\left(\frac{\{g(\omega_j) - p_j\} - \{g(\omega_q) - p_q\} + x}{\mu} + \eta\right)\right\}\right\}.\end{aligned} \tag{173}$$

ここで，

$$\delta = \exp\left\{-\left(\frac{x}{\mu} + \eta\right)\right\}, \tag{174}$$

$$y_q = \exp\left\{\frac{g(\omega_q) - p_q}{\mu}\right\} \quad (q = 0, 1, \cdots, N) \tag{175}$$

とおくと，（164）式は，

$$f(x) = \frac{1}{\mu} \delta \exp\{-\delta\}, \tag{176}$$

また（173）式は，

$$F(\{g(\omega_j) - p_j\} - \{g(\omega_q) - p_q\} + x) = \exp\left\{-\delta \frac{y_q}{y_j}\right\} \tag{177}$$

と表される。（174）式より，

$$d\delta = -\frac{\delta}{\mu} dx \tag{178}$$

であり，x が$-\infty$から∞まで変化するとき，δは0から∞まで変化する。以上を踏まえると，（172）式は次のように計算される。

$$\begin{aligned}\sigma_j &= \int_0^\infty \exp\left\{-\delta \frac{y_j}{y_j}\right\} \prod_{q \neq j} \left(\exp\left\{-\delta \frac{y_q}{y_j}\right\} \right) d\delta \\ &= \int_0^\infty \exp\left\{-\delta \frac{\sum_{q=0}^{N} y_q}{y_j}\right\} d\delta \\ &= \frac{y_i}{\sum_{q=0}^{N} y_q} \\ &= \frac{\exp\{g(\omega_j) - p_j\}}{\sum_{q=0}^{N} \exp\{g(\omega_q) - p_q\}}.\end{aligned} \tag{179}$$

$g(\omega_0) = p_0 = 0$より $\exp\{g(\omega_0) - p_0\} = 1$であるから，（179）式は，

$$\sigma_j = \frac{\exp\{g(\omega_j) - p_j\}}{1 + \sum_{q=1}^{N} \exp\{g(\omega_q) - p_q\}} \tag{180}$$

と書ける。これより，σ_j は$(p_1, \cdots, p_N)$の関数であることが明らかになり，$\sigma_j = \sigma_j(p_1, \cdots, p_N)$と表わすことができる。さらに，

$$\sum_{j=0}^{N} \sigma_j = 1 \tag{181}$$

となるから，財0～財 N の購入確率を足し合わせると 1 になる。このことから，σ_j を企業 j の市場シェアと解釈することができる。

どの企業の限界費用も mc であり，生産量について一定である。また，固定費用はないものとする。企業 j の利潤は，

$$\pi_j = (p_j - mc)M\sigma_j(p_1, \cdots, p_N) \tag{182}$$

となり，利潤最大化の１階の条件は次のようになる。

$$\frac{\partial \pi_j}{\partial p_j} = M\sigma_j + (p_j - mc)M\frac{\partial \sigma_j}{\partial p_j} = 0\cdot \tag{183}$$

ここで，

$$\frac{\partial \sigma_j}{\partial p_j} = -\sigma_j + \sigma_j{}^2 \tag{184}$$

であるから，１階の条件は，

$$-(p_j - mc)(1 - \sigma_j) + 1 = 0 \tag{185}$$

となる。すべての $j = 1, \cdots, N$ についてこの式を満たす $\boldsymbol{p} = (p_1, \cdots, p_N)$が最適価格である。つまり，$\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_N)$，$\boldsymbol{i} = (1, \cdots, 1)$として，$\boldsymbol{p}$ の連立方程式

$$-(\boldsymbol{p} - mc\times\boldsymbol{i})\circ\{\boldsymbol{i} - \boldsymbol{\sigma}(\boldsymbol{p})\} + \boldsymbol{i} = \boldsymbol{o} \tag{186}$$

を解くことで最適価格を得る[75]。

数値計算を行うために，g を次のように特定化する。

$$\exp\{g(\omega)\} = \begin{cases} \exp\{\omega\} & （\omega \leq \omega^* のとき） \\ \exp\{\omega^*\}\{2 - \exp\{\omega^* - \omega\}\} & （\omega > \omega^* のとき） \end{cases} \tag{187}$$

定数ω^*は，２つの効用曲線をつなぎ合わせる屈曲点（point of curvature）である（図３）[76), 77)]。

---

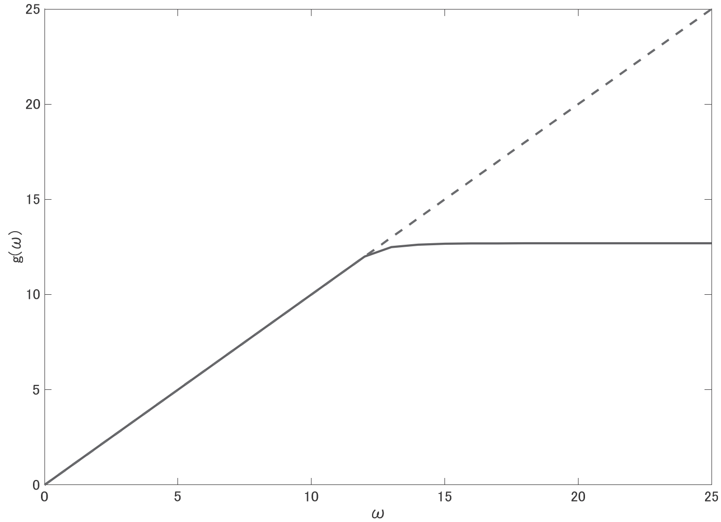
75) ○はアダマール積を表す。MATLAB®では .* 演算子で計算できる。

76) ２つの関数 $g_1(\omega)$，$g_2(\omega)$について，屈曲点ω^*では $g_1(\omega^*) = g_2(\omega^*)$かつ $g_1'(\omega^*) = g_2'(\omega^*)$となる。

77) ω^*は g を有界関数にして，なおかつスムーズな関数にする。実際

$$g(\omega) \to \omega^* + \log 2 \quad (\omega \to \infty)$$

となるから，g は上に有界となる。つまり，ωが十分大きな値にあるとき，ωが増加しても g はほとんど増加しない。g がωについて上に有界であるとき，利潤もωについて上に有界となる。これは，投資を実行してωを上昇させても需要量はほとんど増加せず，したがって，利潤もほとんど増加しないことを意味する。ゆえに，高い水準の効率性を持つ企業は投資をするインセンティブがないので，ωはそれ以上上昇しない。このことから，実現可能な効率性パラメータの集合は有限集合になる。



$\omega^* = 12$とする
破線はωのグラフ

図３　g (ω)のグラフ

6.2.2　コード

```
function [] = cqprofit(nfirms, descn)
  global w p M sigma mc profstar pstar
  global profit agprof share pmcmarg concent

  i = 1;
  while i <= descn
    w = qdecode(i);
    w = 3*w - 7;
    pstar = newton(p, 'cfunk');
    profstar = M*pstar.*sigma - M*mc*sigma;
    profit(i, :) = profstar';
```

```
12        share(i, :) = sigma';
13        pmcmarg(i) = sum(pstar.*sigma)/mc/sum(sigma);
14        concent(i) = max(sigma)/sum(sigma);
15        i = i+1;
16      end
17    end
```

エンコードされた産業構造 i = 1, …, descn の下，nfirms 社の企業の間でベルトラン競争が展開される。各企業が得る均衡利潤を計算して，利潤行列の第 i 行に格納する。また，市場シェア，利潤マージン，集中度といった統計量も同時に計算する。

まず，産業構造 i をデコードして w を得る（07行目）。さらに，w を3w －7に変換する（08行目）[78]。関数 cfunk は，利潤最大化の１階の条件（186）式の左辺を，$\boldsymbol{p}$ の関数と見たものである。関数 cfunk の根が最適価格 pstar である。この根をニュートン＝ラフソン法で求める（09行目）。財の購入確率 sigma は関数 cfunk 内で計算される。そして，sigma と pstar を用いて均衡利潤 profstar を計算する（10行目）。これを利潤行列 profit の第 i 行に格納する（11行目）。sigma は各企業の市場シェアを表すので，これを市場シェア行列 share の第 i 行に格納する（12行目）。さらに，産業構造 i での利潤マージン

$$\mathrm{pmcmarg} = \frac{\sum_{n=1}^{N} p_n \sigma_n}{mc \sum_{n=1}^{N} \sigma_n} \tag{188}$$

を計算して，利潤マージン・ベクトル pmcmarg の第 i 要素に格納する（13行目）。また，１社集中度

78) つまり，$\omega_n \in \{0, 1, 2, 3, \cdots\}$を$\omega_n \in \{-7, -4, -1, 2, \cdots\}$に変換して需要量を計算する。

$$\texttt{concent} = \frac{\mathop{\mathrm{Max}}\limits_{n} \sigma_n}{\sum_{n=1}^{N} \sigma_n} \tag{189}$$

を計算して，１社集中度ベクトル concent の第 i 要素に格納する（14行目）。

```
01  function [ret] = cfunk(p)
02    global w mc sigma egw
03    egw = eg(w);
04    n = egw.*exp(-p);
05    denom = 1.0 + sum(n);
06    sigma = n ./ denom;
07    ret = -(p-mc).*(1-sigma) + 1;
08  end
```

関数 cfunk は，入力された nfirms 次元の価格ベクトル p を（186）式の左辺に代入して，その値を出力する。この値をゼロとする p が最適価格となる。産業構造 w = ($\omega_1, \cdots, \omega_{\mathtt{nfirms}}$)は関数 cqprofit で変換されたものを使用するので，グローバル変数として共有する（02行目）。関数 eg を呼び出して egw = ($\exp\{g(\omega_1)\}, \cdots, \exp\{g(\omega_{\mathtt{nfirms}})\}$)（03行目），さらに，n = ($\exp\{g(\omega_1) - p_1\}, \cdots, \exp\{g(\omega_{\mathtt{nfirms}}) - p_{\mathtt{nfirms}}\}$)（04行目）と計算していく。（180）式について分母 denom（05行目），sigma（06行目）をそれぞれ計算する。こうして（186）式左辺の値が計算されたので，これを戻り値とする（07行目）。

```
function [wret] = eg(w)
  global wstar
  wret = zeros(size(w,1), 1);

  i = 1;
```

```
06     while i <= size(w, 1)
07       if w(i) <= wstar
08         wret(i) = exp(w(i));
09       else
10         wret(i) = exp(wstar)*(2.0-exp(-(w(i)-wstar)));
11       end
12
13       i = i+1;
14     end
15   end
```

入力された産業構造 w = $(\omega_1, \cdots, \omega_{\mathtt{nfirms}})$に対して$\exp\{g(\boldsymbol{\omega})\} = (\exp\{g(\omega_1)\}, \cdots, \exp\{g(\omega_{\mathtt{nfirms}})\})$を計算して，計算結果 wret を出力する関数である。wstar はω^*に相当し，ω_{i}（i = 1, …, nfirms）が wstar よりも小さいとき（08行目）と，大きいとき（10行目）に分けて，それぞれ$\exp\{g(\omega_{\mathrm{i}})\}$を計算する。これを wret の第 i 要素に格納する。

```
function [out1] = newton(p, objfunk)
  id = eye(size(p, 1));
  epsilon = 0.0001;
  tol = 1e-8;
  maxiter = 100;
  iter = 0;
  norm = tol+1;
  deriv = zeros(size(p, 1), size(p, 1));

  while (norm > tol) && (iter < maxiter)
    iter = iter+1;
    [x] = feval(objfunk, p);
```

```

    i = 1;
    while i <= size(p, 1)
      [y] = feval(objfunk, p + epsilon.*id(:, i));
      deriv(:, i) = (y - x)/epsilon;
      i = i+1;
    end

    pnew = p - 0.6*inv(deriv)*x;
    norm = max(abs(pnew - p));
    p = pnew;
  end

  if norm > tol
    %ニュートン=ラフソン法失敗%
  end

  out1 = pnew;
end
```

関数 objfunk と初期点 p $=(p_1, \cdots, p_N)$を入力して，ニュートン＝ラフソン法により根 pnew を求めて出力する。

k 回反復後の根の推測値を $p^{(k)}$とする。関数 objfunk の点($p^{(k)}$, objfunk($p^{(k)}$))における線形近似式は，

$$\mathrm{objfunk}(p) = \mathrm{objfunk}(p^{(k)}) + \frac{\partial \mathrm{objfunk}}{\partial p}(p^{(k)})(p - p^{(k)})\,. \tag{190}$$

ただし，偏微分係数行列

$$\frac{\partial \texttt{objfunk}}{\partial \mathrm{p}} = \begin{pmatrix} \frac{\partial \texttt{objfunk}}{\partial \mathrm{p}_1} & \cdots & \frac{\partial \texttt{objfunk}}{\partial \mathrm{p}_N} \\ \vdots & & \vdots \\ \frac{\partial \texttt{objfunk}}{\partial \mathrm{p}_1} & \cdots & \frac{\partial \texttt{objfunk}}{\partial \mathrm{p}_N} \end{pmatrix}. \tag{191}$$

根では objfunk(p) = $\boldsymbol{o}$ となるから，改良された根の推測値 $\mathrm{p}^{(k+1)}$は，

$$\texttt{objfunk}(\mathrm{p}^{(k)}) + \frac{\partial \texttt{objfunk}}{\partial \mathrm{p}}(\mathrm{p}^{(k)})(\mathrm{p}^{(k+1)} - \mathrm{p}^{(k)}) = \boldsymbol{o} \tag{192}$$

を満たさなければならない。偏微分係数行列の逆行列が存在するならば，

$$\mathrm{p}^{(k+1)} = \mathrm{p}^{(k)} - \texttt{objfunk}(\mathrm{p}^{(k)})\left[\frac{\partial \texttt{objfunk}}{\partial \mathrm{p}}(\mathrm{p}^{(k)})\right]^{-1} \tag{193}$$

となる。ここでは，根に収束しない事態を回避するため，また収束の速度を向上させるために，偏微分係数行列の逆行列に0.6をかけて計算する（図 4 参照）[79]。偏微分係数は次のようにして計算する。

$$\frac{\partial \texttt{objfunk}}{\partial \mathrm{p}_i} = \frac{\texttt{objfunk}(\mathrm{p}_1, \cdots, \mathrm{p}_i + \varepsilon, \cdots, \mathrm{p}_N)}{(\mathrm{p}_i + \varepsilon) - \mathrm{p}_i}. \tag{194}$$

コードは次のようになっている。（194）式のεを0.0001とする（03行目）。許容誤差 tol，および反復回数の上限 maxiter を定める（04，05行目）。反復回数 iter，および誤差 norm を初期化する（06，07行目）。誤差が許容誤差 tol を超えており，かつ反復回数 iter が maxiter 回を超えない限り，ニュートン＝ラフソン法を繰り返して根の推測値を改良する。

ニュートン＝ラフソン法では次のような処理が行われる。x = objfunk(p)，y = objfunk($\mathrm{p}_1, \cdots, \mathrm{p}_i + \varepsilon, \cdots, \mathrm{p}_N$)の値をそれぞれ求めて（12，16行目），（194）式に従って偏微分係数を計算した上で，deriv の第 i 列に格納する（17行目）。これを $i = 1, \cdots, N$ について計算することで，

79) 減衰（damped）ニュートン＝ラフソン法と呼ばれる。0.6 は減衰係数である。

$$\mathtt{deriv} = \frac{\partial \mathtt{objfunk}}{\partial \mathtt{p}} \tag{195}$$

を得る。(193) 式に従い，改良された根の推測値 pnew を求める（21行目)。そして，誤差

$$\begin{aligned}\mathtt{norm} &= \left\| \mathtt{p}^{(k+1)} - \mathtt{p}^{(k)} \right\| \\ &= \operatorname*{Max}_{i} \left| \mathtt{pnew} - \mathtt{p} \right| \end{aligned} \tag{196}$$

を計算する（22行目)。pnew を p に代入して，ニュートン＝ラフソン法を繰り返す（23行目)。

反復回数 iter が maxiter 回を超えたことで **while** ループを抜けたとき，それでもなお誤差が tol を上回っていたら，根は収束しなかったと見なし，ニュートン＝ラフソン法は失敗したと判断する（26〜28行目)。そうでないときは収束したと判断して，pnew を objfunk の根として値を返す（30行目)。

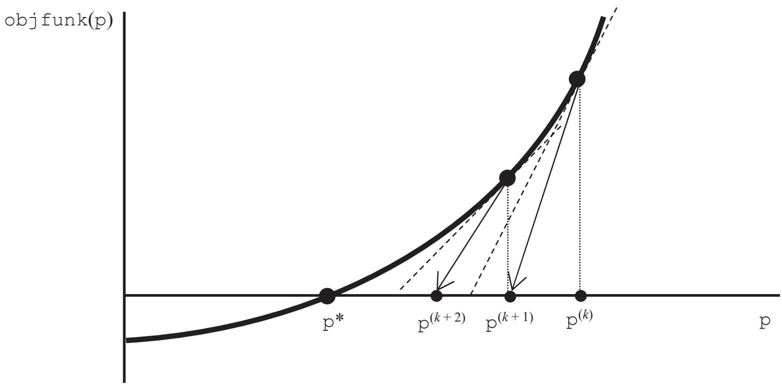


図４　減衰ニュートン＝ラフソン法

7．シミュレーション

7.1 コード

```
wthis = [entry_k+2; zeros(rlnfirms-1, 1)];
numtimes = 10000;

load(['a.' PREFIX '_markov' int2str(rlnfirms) '.mat']);

lifedis = [];
valuedis = [];
lifemx = zeros(rlnfirms, 1);
valuemx = zeros(rlnfirms, 1);

t = 0;
while t < numtimes
  codew2 = encode(wthis);
  pr = prising(codew2, :)';
  xx = newx(codew2, :)';
  vv = newvalue(codew2, :)';

  wtrans = zeros(rlnfirms, 1);
  [~, vind] = min(vv);
  i = (min(vv) == phi)*(vind-1) + (min(vv) > phi)*rlnfirms;

  if i > 0
    wtrans(1:i) = wthis(1:i);
  end

  lifemx = lifemx + (wthis > 0);
```

```
thisexit = (wtrans == 0) && (lifemx > 0);
lifemx = lifemx - (wthis > 0);
if sum(thisexit) > 0
  [~, ~, v1] = find(lifemx.*thisexit);
  [~, ~, v2] = find(valuemx.*thisexit);
  lifedis = [lifedis; v1];
  valval = v2 + (beta.^(1+v1))*phi;
  valuedis = [valuedis; valval];
  lifemx = lifemx .* (1-thisexit);
  valuemx = valuemx .* (1-thisexit);
end
codew = encode(wtrans);
valuemx = valuemx
+ (beta.^lifemx) .* (-xx + profit(codew, :)');
yesentry = 0;
entrypr = rand(1, 1);
yesentry = (isentry(codew) > entrypr);
if yesentry
  wtrans(rlnfirms) = entry_k;
  entryfee = x_entryl + entrypr * (x_entryh - x_entryl);
  valuemx(rlnfirms) = -entryfee;
else
  entryfee = 0;
```

```
        end

        lifemx = lifemx + (wtrans > 0);
        wnext = wtrans
        + (pr >= rand(rlnfirms, 1)) - (rand(1, 1) <= delta);
        wnext = max(([wnext, zeros(rlnfirms, 1)])')';
        temp = flipud(sortrows([wnext, lifemx, valuemx], 1));
        wthis = temp(:, 1);
        lifemx = temp(:, 2);
        valuemx = temp(:, 3);
        t = t+1;
    end
```

５節で計算されたマルコフ完全ナッシュ均衡をもとに，産業動学のシミュレーションを行う。メインルーチンの実行により，均衡が生成する価値関数 newvalue，投資関数 newx，効率性の上昇確率 prising，参入確率 isentry といったデータがファイルに保存されている。シミュレーションでは，このファイルをロードしてデータをモデルに与え，企業行動と産業構造の動態を観察する。データ newvalue, newx, prising は行列であり，行番号はエンコードされた産業構造，列番号は企業のインデックスに対応する。毎期，どの企業も newx に従って投資を実行し，newvalue がスクラップ価値 phi よりも低くなると退出する。

毎期間［0, 1］上の一様分布から rlnfirms 個の値をランダムに抽出して，各企業に値を割り振る。prising よりも小さい値を割り振られた企業は，効率性パラメータが上昇したと解釈する[80)]。isentry はベクトルであり，要素のインデックスはエンコードされた産業構造に対応する。毎期間［0, 1］上の一様分布から値をランダムに抽出して，それが isentry よりも小さい値ならば，参入が

80) ランダムに抽出した値が[0, prising]に収まるならば，効率性は上昇する。したがって，投資水準が高く，その結果 prising の値が大きくなっている企業ほど，効率性が上昇する可能性が高まる。

発生したと解釈する。

初期の産業構造を外生的に与えて，10,000期のシミュレーションを行う。10,000期間において産業で活動したすべての企業について，生存期間 lifemx と獲得した総利潤の割引現在価値 valuemx を記録して，それぞれの分布 lifedis，valuedis を導き出す。

以下では，コードを解説していく。

01行目：初期の産業構造を wthis $=(\mathscr{o}+2, 0, \cdots, 0)$とする。

02行目：シミュレーションの期間 numtimes を設定する。0期から numtimes－1期までシミュレーションを行う。

04行目：マルコフ完全ナッシュ均衡値が保存されているファイルから，データ newvalue，newx，prising，isentry を読み込む。

06～09行目：lifedis，valuedis，lifemx，valuemx を初期化する。

13行目：産業構造 wthis をエンコードして codew2を得る。

14行目：産業構造 codew2における効率性の上昇確率 pr を求める。

15行目：産業構造 codew2における投資関数ベクトル xx を求める。

16行目：産業構造 codew2における価値関数ベクトル vv を求める。

18行目：退出が完了した後の産業構造 wtrans について，これを初期化する。

19行目：価値関数ベクトル vv を構成する rlnfirms 個の要素の中で最小値を見つけ，その要素のインデックスを vind とする。

20行目：事業を継続する企業数 i を求める。vv の最小値がスクラップ価値 phi に等しいならば，vind－1社が事業を継続する。vv の最小値が phi よりも大きいならば退出は起こらず，rlnfirms 社すべてが事業を継続する。

22～24行目：退出が完了した後の産業構造 wtrans を構築する。企業数 i が正ならば，事業を継続する企業が存在する。よって，1～i 番目の企業は産業で活動しているので，産業構造 wthis の第1～i 要素を wtrans の第1～i 要素に代入する。i＋1～rlnfirms 番目の企業は退出しているので，wtrans の第 i＋1～rlnfirms 要素については，初期化を維持して値をゼロとする。

26行目：wthis の要素が正である企業，すなわち，効率性が正である企業はこの期には生存しているので，これらの企業の生存期間 lifemx を1つ増やす。

27行目：lifemx の要素が正であり，なおかつ wtrans の要素がゼロである企業，すなわち，今期は生存しており，かつこの期に退出する企業を考える。これら企業のインデックスに対応する thisexit の要素を 1 と置いて目印を付ける。
28行目：26行目で変更した lifemx を元に戻す。つまり，lifemx は前期までの生存期間である。
30行目：thisexit の要素の和が正であるとき，退出する企業が存在する。これらの企業について，生存期間を lifedis に，これまで獲得した総利潤の割引現在価値を valuedis に，それぞれ記録する。31～36行目でこの処理を行う。
31行目：退出する企業のインデックスに対する要素にはその企業の lifemx を格納し，残りはゼロを置いたものが，ベクトル lifemx.*thisexit である。したがって，v1は退出する企業の lifemx を一列に並べたベクトルである[81]。
32行目：31行目と同様に，v2は退出する企業の valuemx を一列に並べたベクトルである。
34行目：lifedis に v1を追加する。退出する企業が現れるたびに，これらの企業の生存期間を lifedis に追加していく。
35行目：退出する企業はこれまで合計 v2の利潤を得ている。そして，参入してから 1＋ v1期後に退出してスクラップ価値 phi を得るので，割り引いた価値を v2に足し合わせる。これが valval であり，退出する企業がこれまで獲得した総利潤の割引現在価値となる。
36行目：valuedis に valval を追加する。退出する企業が現れるたびに valval を計算して，この値を valuedis に追加していく。
38，39行目：退出する企業の lifemx と valuemx をゼロにリセットし，それ以外の企業の lifemx と valuemx はそのままにしたものを，lifemx や valuemx に置き換える。
42行目：退出が完了した後の産業構造 wtrans をエンコードして codew を得る。
43行目：今期も事業を継続する企業について，彼らがこれまで獲得してきた総利潤の割引現在価値を計算する。前期まで獲得した総利潤の割引現在価値 valuemx

81) [~, ~, v] ＝ find(A)は，行列 A の非ゼロ要素を v へ一列に格納する MATLAB®の組み込み関数である。

に，今期獲得した純利益を割り引いた上で足し合わせる。これを改めて valuemx に置き換える。純利益は，産業構造 codew で実現する利潤から，産業構造 codew2 で実行される投資額を差し引いたものである[82]。この純利益は，企業が参入してから lifemx 期後に獲得するので，割引率 $\beta^{\mathtt{lifemx}}$ で割り引く。

45行目：yesentry を 0 に初期化する。

46行目：[0, 1] 上の一様分布から値をランダムに抽出し，これを entrypr とする。

47行目：産業構造 codew における参入確率 isentry が entrypr よりも大きいときは，参入が発生したと見なして，yesentry を 1 とする[83]。

49行目：yesentry が 1 であるときは参入が発生するので，50～52行目で参入が発生した後の産業構造を構築し，参入企業の価値を計算する。

50行目：wtrans の rlnfirms 番目の要素に entry_k を代入して，参入が発生した後の産業構造 wtrans を構築する。

51行目：参入確率 entrypr に対応する参入費用 entryfee を計算する。参入費用 x^e は，[x_entryl, x_entryh]上の一様分布に従う確率変数であるから，次式が成り立つ[84]。

$$\mathtt{entrypr} = \Pr(x^e \le \mathtt{entryfee}) = \frac{\mathtt{entryfee} - \mathtt{x\_entryl}}{\mathtt{x\_entryh} - \mathtt{x\_entryl}}. \qquad (197)$$

52行目：rlnfirms 番目の企業は新規参入企業であり，今期は参入費用 entryfee を負担する。

53, 54行目：yesentry が 0 であるときは参入が発生しないので，参入費用をゼ

82) 事業継続の価値関数における利潤は，退出が完了する前の産業構造 $\boldsymbol{\omega}$（エンコードされた産業構造は codew2）で実現する利潤で評価する（(8), (71)式）。この価値関数をスクラップ価値と比較して退出を決める。他方，事業を継続するときに実際に獲得する利潤は，退出が起こった後の産業構造 $\boldsymbol{\omega}^{\dagger}$（エンコードされた産業構造は codew）で実現する利潤になる。

83) 産業構造に空きスロットがないときは，isentry がゼロとなるので，どのような entrypr が抽出されても参入は起こらない。

84) 参入価値 V^e について，$V^e \ge \mathtt{entryfee}$ のとき参入が起こる。これを確率で評価すると，$\Pr(x^e \le V^e) \ge \Pr(x^e \le \mathtt{entryfee})$ となり，これが $\mathtt{isentry} \ge \mathtt{entrypr}$ に対応する。

ロとする。

57行目：産業構造 `wthis` において，効率性が正である企業の生存期間 `lifemx` を１つ増やす。すなわち，28行目の処理で `lifemx` を前期までの生存期間としたので，今期の生存を `lifemx` に反映させる。

58行目：各企業の効率性が変化した後の産業構造 `wnext` を構築する。[0, 1] 上の一様分布から `rlnfirms` 個の値をランダムに抽出する。`rlnfirms` 個の値を並べたベクトルと確率ベクトル `pr` を，要素ごとに比較する。前者の値が小さくなる要素について，そのインデックスに対応する企業は効率性が上昇したものと判定する。また，[0, 1] 上の一様分布から値を１つ抽出する。この値が確率 `delta` よりも小さくなるならば，産業全体にショックが生じて，すべての企業の効率性は低下したものと判定する。

59行目：`wnext` の要素が０未満になるものは０に調整する。

60～63行目：`wnext` の要素を降順に並べ替えて `wthis` とする。`wthis` が次期の産業構造となる。`wnext` の並び替えに合わせて，`lifemx` と `valuemx` も同じように並べ替える。

64行目：次の期に進む。

7.2　投資関数と価値関数のグラフ

マルコフ完全ナッシュ均衡によって生成される企業１と企業２の投資水準と価値のグラフを，平面 (ω_1, ω_2)，$\omega_1 \geq \omega_2$上に描く（図５～12）[85]。パラメータを次のように置く。

$N = 2$，$x_H^e = 0.25$，$x_L^e = 0.15$，$\omega^e = 4$，$\beta = 0.925$，$\delta = 0.7$，$\phi = 0.1$，$a = 3$，$mc = 5$，$M = 5$，$\overline{\omega} = 25$，$\omega^* = 12$，$D = 3$，$f = 0.2$，$\gamma = 1$.

85) データ点の間を線形で補間してグラフを描いている。

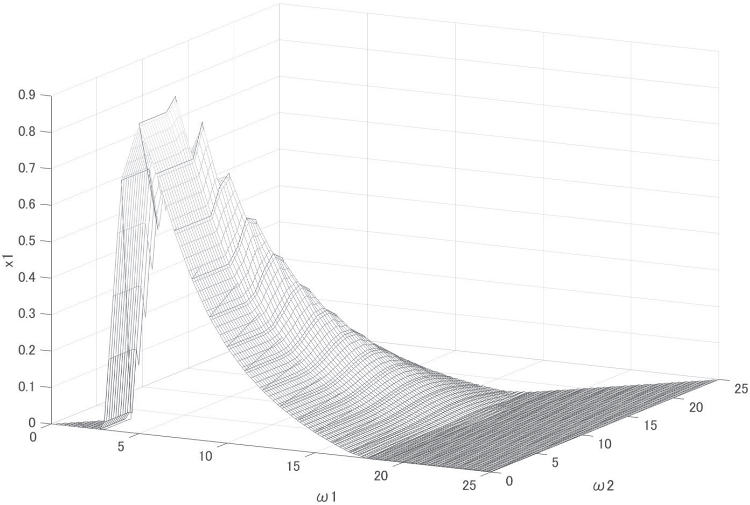


図５　クールノー競争における企業１の投資関数のグラフ

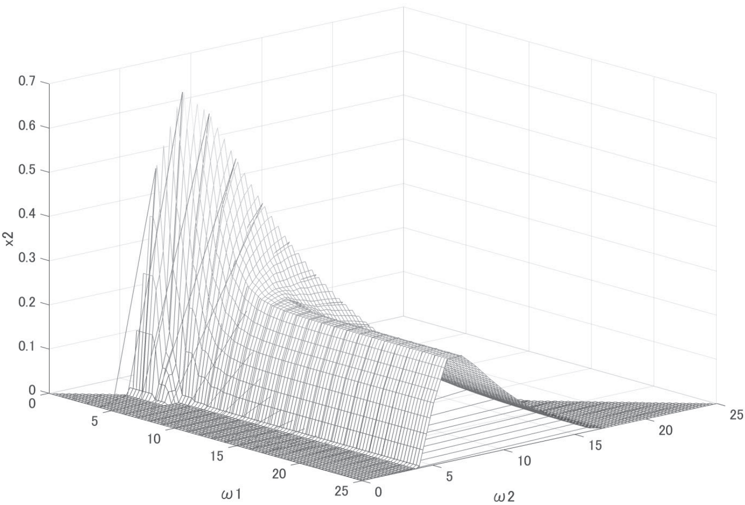


図６　クールノー競争における企業２の投資関数のグラフ

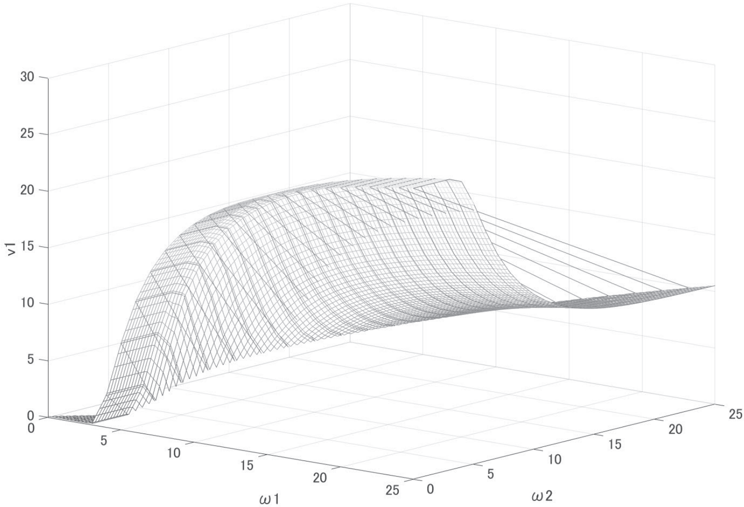


図７　クールノー競争における企業１の価値関数のグラフ

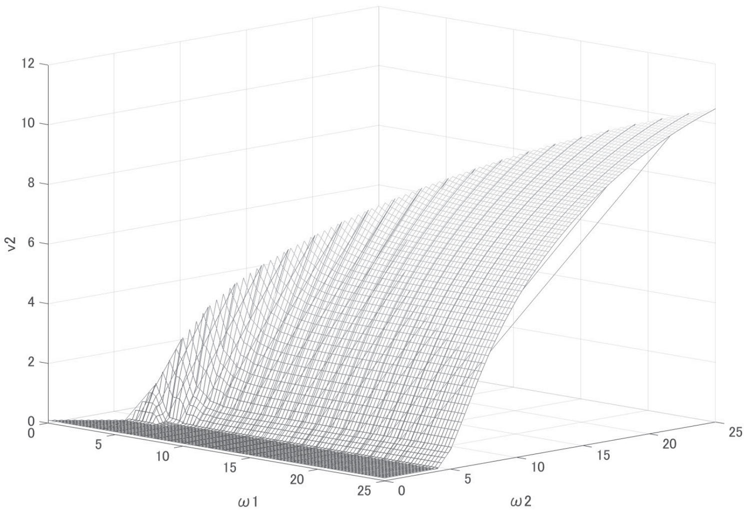


図８　クールノー競争における企業２の価値関数のグラフ

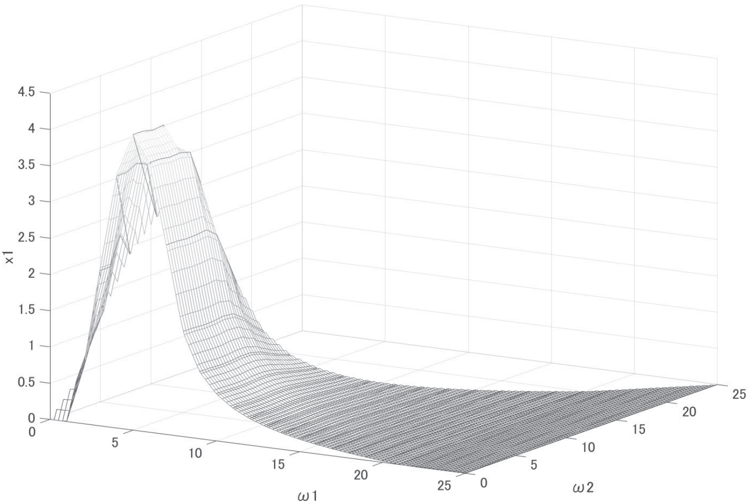


図９　ベルトラン競争における企業１の投資関数のグラフ

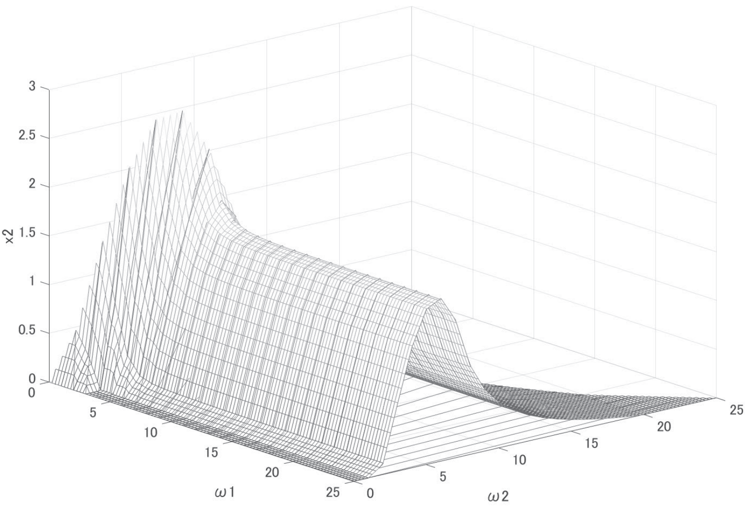


図10　ベルトラン競争における企業２の投資関数のグラフ

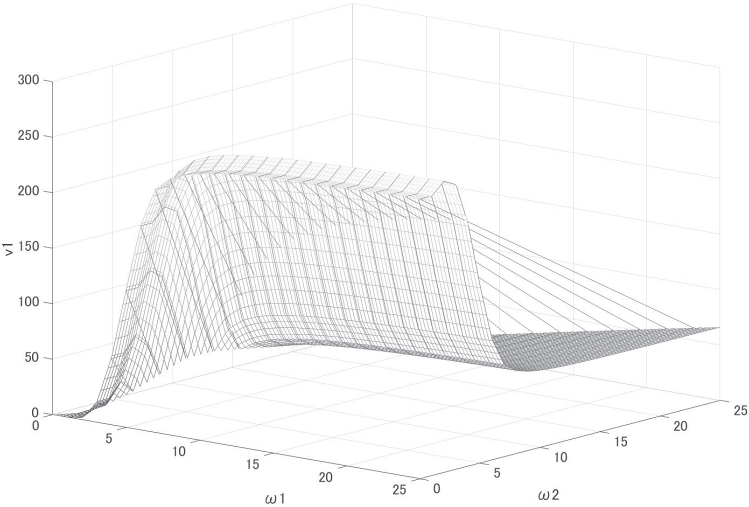


図11　ベルトラン競争における企業１の価値関数のグラフ

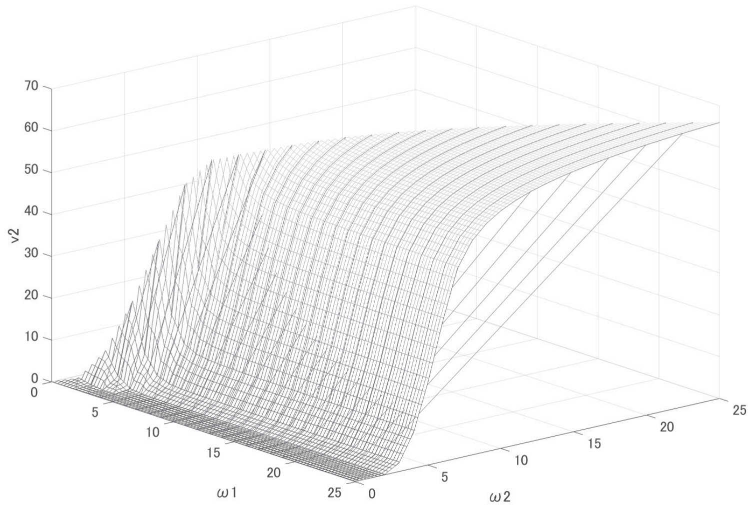


図12　ベルトラン競争における企業２の価値関数のグラフ

7.3　シミュレーションの結果

マルコフ完全ナッシュ均衡によって導き出される産業動学を，シミュレーションにより確認する。３社の企業の間でクールノー競争，もしくはベルトラン競争が10,000期間繰り返される。$N=3$以外のパラメータは，前節と同じ数値を設定する。シミュレーションの結果を表７および表８に示す[86)]。

（続く）

企業が存在しない期間：1,353
１社が活動する期間：7,990
２社が活動する期間：657
３社が活動する期間：０
退出が発生する期間：2,269
参入が発生する期間：2,270
退出と参入が同時に発生する期間：1,647
10,000期間の平均投資量：0.59（0.29）
10,000期間の平均利潤マージン：37.99（150.52）
10,000期間の平均１社集中度：0.84（0.35）
価値の分布 `valuedis` の平均：0.38（2.02）
生存期間の分布 `lifedis` の平均：5.10（17.33）

（　）の値は標準偏差

表７　シミュレーションの結果（クールノー競争）

企業が存在しない期間：０
１社が活動する期間：11
２社が活動する期間：9,098
３社が活動する期間：891
退出が発生する期間：226
参入が発生する期間：228
退出と参入が同時に発生する期間：127
10,000期間の平均投資量：1.94（0.97）
10,000期間の平均利潤マージン：1.44（0.15）
10,000期間の平均１社集中度：0.54（0.11）
価値の分布 `valuedis` の平均：1.20（10.67）
生存期間の分布 `lifedis` の平均：90.52（363.73）

（　）の値は標準偏差

表８　シミュレーションの結果（ベルトラン競争）

---

86) 効率性の上昇・下落，あるいは，参入の発生は確率的に決まるので，シミュレーションを実施するたびに結果は微妙に異なる。表７・表８は結果の１つに過ぎない。

参考文献

[1] 伊里正夫，藤野和健(1985)『数値計算の常識』，共立出版。
[2] 奥村晴彦(1991)『C 言語による最新アルゴリズム事典』，技術評論社。
[3] 皆本晃弥(2005)『C 言語による数値計算入門：解法・アルゴリズム・プログラム』，サイエンス社。
[4] 蓑谷千凰彦(1998)『すぐに役立つ統計分布』，東京図書。
[5] Anderson, S., A.de Palma., and J-F.Thisse.(1992), *Discrete Choice Theory of Product Differentiation*, MIT Press.
[6] Doraszelski, U. and A. Pakes. (2007), “A Framework for Applied Dynamic Analysis in IO”, In Armstrong, M. and R.Porter.(eds.), *Handbook of Industrial Organization* : *Volume 3*, North-Holland, pp.1887-1966.
[7] Judd, K. (1998), *Numerical Methods in Economics*, MIT Press.
[8] Mathews, J. and K. Fink. (2003), *Numerical Methods Using MATLAB*, Pearson Prentice-Hall.
[9] Miranda, M. and P. Fackler. (2002), *Applied Computational Economics and Finance*, MIT Press.
[10] Pakes, A., G. Gowrisankaran., and P. McGuire. (1993), “Implementing the Pakes-McGuire Algorithm for Computing Markov Perfect Equilibria in Gauss”, Working Paper, Yale University.
[11] Pakes, A. and P. McGuire. (1993), “Computing Markov-Perfect Nash Equilibria : Numerical Implications of a Dynamic Differentiated Product Model.”, *RAND Journal of Economics*, Vol.25, pp.555-589.
[12] Press, W., S. Teukolsky., W. Vetterling., and B.Flannery.(2007), *Numerical Recipes* : *The Art of Scienctific Computing*, Cambridge University Press.
[13] Stachurski, J. (2009), *Economic Dynamics* : *Theory and Computation*, MIT Press.
[14] Train, K. (2009), *Discrete Choice Methods with Simulation*, Cambridge University Press.