Solving Dynamic Games with Newton's Method

Michael Ferris[1] Kenneth L. Judd[2] Karl Schmedders[3]

[1]Department of Computer Science, University of Wisconsin at Madison

[2]Hoover Institution, Stanford University

[3]Dept. of Business Administration, Univ. of Zurich and Swiss Finance Institute

Zurich Initiative on Computational Economics

September 5, 2011

swiss:finance:institute

Discrete-Time Finite-State Stochastic Games

Central tool in analysis of strategic interactions among forward-looking players in dynamic environments

Example: The Ericson & Pakes (1995) model of dynamic competition in an oligopolistic industry

Little analytical tractability

Most popular tool in the analysis: The Pakes & McGuire (1994) algorithm to solve numerically for an MPE (and variants thereof)

Applications

Advertising (Doraszelski & Markovich 2007)

Capacity accumulation (Besanko & Doraszelski 2004, Chen 2005, Ryan 2005, Beresteanu & Ellickson 2005)

Collusion (Fershtman & Pakes 2000, 2005, de Roos 2004)

Consumer learning (Ching 2002)

Firm size distribution (Laincz & Rodrigues 2004)

Learning by doing (Benkard 2004, Besanko, Doraszelski, Kryukov & Satterthwaite 2010)

Applications cont'd

Mergers (Berry & Pakes 1993, Gowrisankaran 1999)

Network externalities (Jenkins, Liu, Matzkin & McFadden 2004, Markovich 2004, Markovich & Moenius 2007)

Productivity growth (Laincz 2005)

R&D (Gowrisankaran & Town 1997, Auerswald 2001, Song 2002, Judd et al. 2011)

Technology adoption (Schivardi & Schneider 2005)

International trade (Erdem & Tybout 2003)

Finance (Goettler, Parlour & Rajan 2004, Kadyrzhanova 2005).

Need for better Computational Techniques

Doraszelski and Pakes (2007)

"Moreover the burden of currently available techniques for computing the equilibria to the models we do know how to analyze is still large enough to be a limiting factor in the analysis of many empirical and theoretical issues of interest."

Purpose of this paper: Solve large models with Newton's Method

Need for better Computational Techniques II

Weintraub et al. (2008)

"There remain, however, some substantial hurdles in the application of EP-type models. Because EP-type models are analytically intractable, analyzing market outcomes is typically done by solving for Markov perfect equilibria (MPE) numerically on a computer, using dynamic programming algorithms (e.g., Pakes and McGuire (1994)). This is a computational problem of the highest order. [...] in practice computational concerns have typically limited the analysis [...] Such limitations have made it difficult to construct realistic empirical models, and application of the EP framework to empirical problems is still quite difficult [...] Furthermore, even where applications have been deemed feasible, model details are often dictated as much by computational concerns as economic ones."

Outline

Cournot Competition

Single good produced by $N = 2$ firms

Firm i's production quantity q_i

Total output $Q = q_1 + q_2$ sold at a single price $P(Q)$

Cost to firm i of producing q_i is $C_i(q_i)$

Firms' profit functions (revenue minus cost)

$$\begin{aligned}\pi_1(q_1, q_2) &= q_1\ P(q_1 + q_2) - C_1(q_1)\\ \pi_2(q_1, q_2) &= q_2\ P(q_1 + q_2) - C_2(q_2)\end{aligned}$$

Dynamic Model

Infinite-horizon game in discrete time $t = 0, 1, 2, \ldots$

At time t firm i is in one of finitely many states, $\theta_{i,t} \in \Theta_i$

State space of the game $\Theta_1 \times \Theta_2$

State of the game: production cost of two firms

Firms engage in Cournot competition in each period t

$$\begin{aligned}
\pi_{1,t} &= q_{1,t}\ P\left(q_{1,t} + q_{2,t}\right) - \theta_{1,t} C_1(q_{1,t}) \\
\pi_{2,t} &= q_{2,t}\ P\left(q_{1,t} + q_{2,t}\right) - \theta_{2,t} C_2(q_{2,t})
\end{aligned}$$

Efficiency of firm i is given by $\theta_{i,t}$

Learning and Investment

Firms' states can change over time

Stochastic transition to state in next period depends on three forces

Learning: current output may lead to lower production cost

Investment: firms can also make investment expenditures to reduce cost

Depreciation: shock to efficiency may increase cost

Dynamic Setting

Each firm can be in one of S states, $j = 1, 2, \dots, S$

State j of firm i determines its efficiency level
$\theta_i = \Theta^{(j-1)/(S-1)}$ for some $\Theta \in (0, 1)$

Total range of efficiency levels $[\Theta, 1]$ for any S

Possible transitions from state j to states $j-1,\ j,\ j+1$ in
next period

Transition probabilities for firm i depend on
- production quantity q_i
- investment effort e_i
- depreciation shock

Transition Probabilities

Probability of successful learning (j to $j+1$), $\psi(q) = \frac{\kappa q}{1+\kappa q}$

Probability of successful investment (j to $j+1$), $\phi(e) = \frac{\alpha e}{1+\alpha e}$

Cost of investment for firm i, $CI_i(e) = \frac{1}{S-1}\left(\frac{1}{2}d_i e^2\right)$

Probability of depreciation shock (j to $j-1$), δ

These individual probabilities, appropriately combined, yield transition probabilities $\Pr\left(\theta' | q, e; \theta\right)$

Transition Probabilities cont'd

Law of motion: State follows a controlled discrete-time, finite-state, first-order Markov process with transition probability

$$\Pr\left((\theta_1', \theta_2')|q_{1,t}, e_{1,t}, q_{2,t}, e_{2,t}; (\theta_{1,t}, \theta_{2,t})\right)$$

Typical assumption of independent transitions:

$$\begin{aligned} & \Pr\left((\theta_1', \theta_2')|q_{1,t}, e_{1,t}, q_{2,t}, e_{2,t}; (\theta_{1,t}, \theta_{2,t})\right) \\ = & \prod_{i=1}^{2} \Pr_i\left(\theta_i'|q_{i,t}, e_{i,t}; \theta_{i,t}\right) \end{aligned}$$

Objective Function

Notation: actions $u_t = (q_{1,t}, e_{1,t}, q_{2,t}, e_{2,t})$, $u_{i,t} = (q_{i,t}, e_{i,t})$
states $\theta_t = (\theta_{1,t}, \theta_{2,t})$

Firm i receives total payoff $\Pi_i(u_t; \theta_t)$ in period t from Cournot competition and investment

Objective is to maximize the expected NPV of future cash flows

$$\mathrm{E}\left\{\sum_{t=0}^{\infty} \beta^t \Pi^i(u_t; \theta_t)\right\}$$

with discount factor $\beta \in (0, 1)$

Bellman Equation

$V_i(\theta)$ is the expected NPV to firm i if the current state is θ

Bellman equation for firm i is

$$V_i\left(\theta\right) = \max_{u_i} \Pi_i\left(u_i, U_{-i}\left(\theta\right); \theta\right) + \beta \mathsf{E}_{\theta'}\left\{V_i\left(\theta'\right) | u_i, U_{-i}\left(\theta\right); \theta\right\}$$

with feedback (Markovian) strategies $U_{-i}\left(\theta\right)$ of other firms

Player i's strategy $U_i(\theta)$ must satisfy

$$U_i(\theta) = \arg\max_{u_i} \left\{\Pi_i\left(u_i, U_{-i}\left(\theta\right); \theta\right) + \beta \mathsf{E}_{\theta'}\left\{V_i\left(\theta'\right) | u_i, U_{-i}\left(\theta\right); \theta\right\}\right\}$$

System of equations defined above for each firm i and each state $\theta \in \Theta$ defines a pure-strategy Markov Perfect Equilibrium

Equilibrium Conditions

Unknowns $U_i(\theta)$, $V_i(\theta)$ for each state θ

$$V_i\left(\theta\right) = \Pi_i\left(u_i, U_{-i}\left(\theta\right);\theta\right) + \beta \mathsf{E}_{\theta'}\left\{V_i\left(\theta'\right)|u_i, U_{-i}\left(\theta\right);\theta\right\}$$

$$\frac{\partial}{\partial u_i}\left\{\Pi_i\left(u_i, U_{-i}\left(\theta\right);\theta\right) + \beta \mathsf{E}_{\theta'}\left\{V_i\left(\theta'\right)|u_i, U_{-i}\left(\theta\right);\theta\right\}\right\} = 0$$

Quadratic cost functions ensure interior solutions $U_i(\theta) >> 0$

First-order conditions are necessary and sufficient

Nonlinear system of equations

Three equations per firm per state, total of $6S^2$ equations

Nonlinear Systems of Equations

System $F(x) = 0$ of n nonlinear equations in n variables
$x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$

$$\begin{aligned}
F_1(x_1, x_2, \ldots, x_n) &= 0 \\
F_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\vdots \\
F_{n-1}(x_1, x_2, \ldots, x_n) &= 0 \\
F_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}$$

Solution Methods

Most popular methods in economics for solving $F(x) = 0$

1. Jacobi Method
   - Value function iteration in dynamic programming
2. Gauss-Seidel Method
   - Iterated best replies in game theory
3. Homotopy Methods
   - Long history in general equilibrium theory
4. Newton's Method
   - Modern implementations largely ignored

Jacobi Method

Last iterate $x^k = (x_1^k, x_2^k, x_3^k, \ldots, x_{n-1}^k, x_n^k)$

New iterate x^{k+1} computed by repeatedly solving one equation in one variable using only values from x^k

$$\begin{aligned}
F_1(x_1^{k+1}, x_2^k, x_3^k, \ldots, x_{n-1}^k, x_n^k) &= 0 \\
F_2(x_1^k, x_2^{k+1}, x_3^k, \ldots, x_{n-1}^k, x_n^k) &= 0 \\
&\vdots \\
F_{n-1}(x_1^k, x_2^k, \ldots, x_{n-2}^k, x_{n-1}^{k+1}, x_n^k) &= 0 \\
F_n(x_1^k, x_2^k, \ldots, x_{n-2}^k, x_{n-1}^k, x_n^{k+1}) &= 0
\end{aligned}$$

Interpretation as iterated simultaneous best reply

Gauss-Seidel Method

Last iterate $x^k = (x_1^k, x_2^k, x_3^k, \ldots, x_{n-1}^k, x_n^k)$

New iterate x^{k+1} computed by repeatedly solving one equation in one variable and immediately updating the iterate

$$
\begin{aligned}
F_1(x_1^{k+1}, x_2^k, x_3^k, \ldots, x_{n-1}^k, x_n^k) &= 0 \\
F_2(x_1^{k+1}, x_2^{k+1}, x_3^k, \ldots, x_{n-1}^k, x_n^k) &= 0 \\
&\vdots \\
F_{n-1}(x_1^{k+1}, x_2^{k+1}, \ldots, x_{n-2}^{k+1}, x_{n-1}^{k+1}, x_n^k) &= 0 \\
F_n(x_1^{k+1}, x_2^{k+1}, \ldots, x_{n-2}^{k+1}, x_{n-1}^{k+1}, x_n^{k+1}) &= 0
\end{aligned}
$$

Interpretation as iterated sequential best reply

Fixed-point Iteration

Reformulation

$$F(x) = 0 \quad \iff \quad x - \alpha F(x) = x$$

yields fixed-point problem $G(x) = x$ with $G(x) = x - \alpha F(x)$

Fixed-point iteration

$$x^{(k+1)} = G(x^{(k)})$$

is also called **Nonlinear Richardson iteration** or **Picard iteration**

Solving a Simple Cournot Game

N firms

Firm i's production quantity q_i

Total output is $Q = q_1 + q_2 + \ldots + q_N$

Linear inverse demand function, $P(Q) = A - Q$

All firms have identical cost functions $C(q) = \frac{2}{3}cq^{3/2}$

Firm i's profit function Π_i is

$$\Pi_i = q_i P(q_i + Q_{-i}) - C(q_i) = q_i (A - (q_i + Q_{-i})) - \frac{2}{3}cq_i^{3/2}$$

First-order Conditions

Necessary and sufficient first-order conditions

$$A - Q_{-i} - 2q_i - c\sqrt{q_i} = 0$$

Firm i's best reply $BR(Q_{-i})$ to a production quantity Q_{-i} of its competitors

$$q_i = BR(Q_{-i}) = \left(\frac{A - Q_i}{2} + \frac{c^2}{8}\right) - \frac{c}{2}\sqrt{\frac{A - Q_{-i}}{2} + \frac{c^2}{16}}$$

Parameter values: $N = 4$ firms, $A = 145$, $c = 4$

Cournot equilibrium $q^i = 25$ for all firms

Jacobi with $N = 4$ firms blows up

$q^0 = (24, 25, 25, 25)$

k	q_1^k	$q_2^k = q_3^k = q_4^k$	$\max_i \lvert q_i^k - q_i^{k-1} \rvert$
1	25	25.4170	1
2	24.4793	24.6527	0.7642
3	25.4344	25.5068	0.9551
4	24.3672	24.3973	1.1095
5	25.7543	25.7669	1.3871
13	29.5606	29.5606	8.1836
14	19.3593	19.3593	10.201
15	32.1252	32.1252	12.766
20	4.8197	4.8197	37.373
21	50.9891	50.9891	46.169

Solving the Cournot Game with Gauss-Seidel

$$q^0 = (10, 10, 10, 10)$$

k	q_1^k	q_2^k	q_3^k	q_4^k	$\max_i \lvert q_i^k - q_i^{k-1} \rvert$
1	56.0294	32.1458	19.1583	11.9263	55.029
2	29.9411	30.8742	25.9424	20.1446	26.088
3	24.1839	26.9767	26.5433	23.8755	5.7571
10	25.0025	25.0016	24.9990	24.9987	5.6080 (-3)
11	25.0003	25.0008	25.0001	24.9995	2.1669 (-3)
12	24.9998	25.0003	25.0002	24.9999	5.8049 (-4)
16	25.0000	25.0000	25.0000	25.0000	1.1577 (-5)
17	25.0000	25.0000	25.0000	25.0000	4.1482 (-6)
18	25.0000	25.0000	25.0000	25.0000	1.1891 (-6)

Contraction Mapping

Let $X \subset \mathbb{R}^n$ and let $G : X \to \mathbb{R}^m$. The function G is **Lipschitz continuous** on X with Lipschitz constant $\gamma \geq 0$ if

$$||G(x) - G(y)|| \leq \gamma ||x - y||$$

for all $x, y \in X$.

Let $X \subset \mathbb{R}^n$ and let $G : X \to \mathbb{R}^n$. The function G is a **contraction mapping** on X if G is Lipschitz continuous on X with Lipschitz constant $\gamma < 1$.

Lipschitz constant of contraction mapping G is also called **modulus** of G

Contraction Mapping Theorem

Contraction Mapping Theorem. Suppose that $G : X \to \mathbb{R}^n$ is a contraction mapping on the closed subset X of $\mathbb{R}^n$ and that $G(X) \subset X$. Then the following conditions hold.

(1) The function G has a unique fixed point $x^* \in X$.

(2) For all $x^{(0)} \in X$ the sequence generated by the fixed-point iteration $x^{(k+1)} = G(x^{(k)})$ converges linearly to x^*. $\square$

Modulus $\gamma < 1$ of G yields constant for linear convergence

$$||x^{(k+1)} - x^*|| = ||G(x^{(k)}) - G(x^*)|| \leq \gamma ||x^{(k)} - x^*||$$

Mode of Updating Iterates

Fixed-point iteration $x^{(k+1)} = G(x^{(k)})$ updates all components of x simultaneously; Jacobi-mode of updating

$$x_i^{(k+1)} = G_i(x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k)}, x_{i+1}^{(k)}, \dots, x_n^{(k)})$$

Gauss-Seidel mode of updating is also possible

$$x_i^{(k+1)} = G_i(x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i^{(k)}, x_{i+1}^{(k)}, \dots, x_n^{(k)})$$

Theorem. Suppose that $G : X \to \mathbb{R}^n$ is a contraction mapping on the set $X = \prod_{i=1}^n X_i$, where each X_i is a nonempty closed subset of $\mathbb{R}$, and that $G(X) \subset X$. Then for all $x^{(0)} \in X$ the sequence generated by the fixed-point iteration $x^{(k+1)} = G(x^{(k)})$ with a Gauss-Seidel mode of updating converges linearly to the unique fixed point x^* of G. $\square$

Alternative Solution Methods

Jacobi component solution method: for all $i = 1, 2, \ldots, n$ the new iterate $x_i^{(k+1)}$ is a solution of the single equation

$$x_i = G_i(x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k)}, \ldots, x_n^{(k)})$$

in the single variable x_i

Gauss-Seidel component solution method: for all $i = 1, 2, \ldots, n$ the new iterate $x_i^{(k+1)}$ is a solution of the single equation

$$x_i = G_i(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \ldots, x_n^{(k)})$$

in the single variable x_i

Convergence of Component Solution Methods

Theorem. Suppose that $G : X \to \mathbb{R}^n$ is a contraction mapping on the set $X = \prod_{i=1}^{n} X_i$, where each X_i is a nonempty closed subset of $\mathbb{R}$, and that $G(X) \subset X$. Then for all $x^{(0)} \in X$ the sequence generated by the Jacobi component solution method converges linearly to the unique fixed point x^* of G. Similarly, the sequence generated by the Gauss-Seidel component solution method converges linearly to x^*. $\square$

Modest generalization to **pseudo-contraction mappings** possible

Sufficient Condition for Contraction Mapping

Theorem. Suppose that X is a nonempty convex subset of $\mathbb{R}^n$ and that $F : X \to \mathbb{R}^n$ is continuously differentiable. Further suppose that

$$\sum_{j \neq i} \left| \frac{\partial F_i(x)}{\partial x_j} \right| < \frac{\partial F_i(x)}{\partial x_i} \leq K$$

for all $i = 1, 2, \ldots, n$ and for all $x \in X$. Then the mapping $G : X \to \mathbb{R}^n$ defined by

$$G(x) = x - \alpha F(x)$$

with $0 < \alpha < \frac{1}{K}$ is a contraction mapping (with respect to the maximum norm). $\square$

Resemblance to a diagonal dominance condition

Iterative Methods for Finding Zeros

SOR = successive overrelaxation

Nonlinear Jacobi SOR method

For all $i = 1, 2, \ldots, n$ solve

$$F_i(x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k)}, \ldots, x_n^{(k)}) = 0$$

for x_i; with $\omega \in (0, 2)$ set

$$x_i^{(k+1)} = x_i^{(k)} + \omega(x_i - x_i^{(k)})$$

Nonlinear Gauss-Seidel SOR method

For all $i = 1, 2, \ldots, n$ solve

$$F_i(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \ldots, x_n^{(k)}) = 0$$

for x_i; with $\omega \in (0, 2)$ set

$$x_i^{(k+1)} = x_i^{(k)} + \omega(x_i - x_i^{(k)})$$

Global Convergence Theorem for Nonlinear SOR Methods

Theorem. Suppose the function $F : \mathbb{R}^n \to \mathbb{R}^n$ has the following properties.

(1) F is a continuous function from $\mathbb{R}^n$ onto $\mathbb{R}^n$.

(2) $F(x) \leq F(y)$ implies $x \leq y$ for all $x, y \in \mathbb{R}^n$.

(3) $F_i : \mathbb{R}^n \to \mathbb{R}$ is decreasing in x_j for all $j \neq i$.

Then for $\omega \in (0, 1]$, any $b \in \mathbb{R}^n$, and from any starting point $x^0 \in \mathbb{R}^n$ the sequences generated by the Jacobi SOR method and the Gauss-Seidel SOR method, respectively, converge to the unique solution x^* of $F(x) = b$. $\square$

Iterates of Jacobi SOR Method, $w = 0.9$

k	q_1^k	$q_2^k = q_3^k = q_4^k$	$\max_i \lvert q_i^k - q_i^{k-1} \rvert$
1	24.9	25.3753	0.9
2	24.5682	24.7937	0.581566
97	27.731	27.731	5.38011
98	22.2193	22.2193	5.51166
99	27.8673	27.8673	5.64804
100	22.0815	22.0815	5.78587
341	43.2918	43.2918	35.6236
342	7.6682	7.6682	35.6236
343	43.2918	43.2918	35.6236
344	7.6682	7.6682	35.6236

Iterates of Jacobi SOR Method, $w = 0.5$

k	q_1^k	$q_2^k = q_3^k = q_4^k$	$\max_i \lvert q_i^k - q_i^{k-1} \rvert$
1	24.5	25.2085	0.5
2	24.6198	25.1215	0.11976
3	24.7339	25.0893	0.11418
4	24.8111	25.0629	0.077200
5	24.8663	25.0446	0.055139
15	24.9957	25.0014	1.7508 (-3)
16	24.9970	25.0010	1.2402 (-3)
17	24.9979	25.0007	8.7845 (-4)
33	25.0000	25.0000	3.5279 (-6)
34	25.0000	25.0000	2.4989 (-6)

Summary

Fixed-point iteration in all its variations (Jacobi mode or Gauss-Seidel mode of updating, Jacobi or Gauss-Seidel component solution method) requires contraction property for convergence

Nonlinear Jacobi SOR or Gauss-Seidel SOR methods require strong monotonicity properties for convergence

Conjecture: these sufficient conditions are rarely satisfied by economic models

Conclusion: do not be surprised if these methods do not work

Methods do have the advantage that they are easy to implement, which explains their popularity in economics

Taylor's Theorem

Theorem. Suppose the function $F : X \to \mathbb{R}^m$ is continuously differentiable on the open set $X \subset \mathbb{R}^n$ and that the Jacobian function J_F is Lipschitz continuous at x with Lipschitz constant $\gamma^l(x)$. Also suppose that for $s \in \mathbb{R}^n$ the line segment $x + \theta s \in X$ for all $\theta \in [0,1]$. Then, the linear function $L(s) = F(x) + J_F(x)s$ satisfies

$$\|F(x+s) - L(s)\| \leq \frac{1}{2}\gamma^L(x)\|s\|^2 \ .$$

$\square$

Taylor's Theorem suggests the approximation
$F(x+s) \approx L(s) = F(x) + J_F(x)s$

Newton's Method in Pure Form

Initial iterate x^0

Given iterate x^k choose Newton step by calculating a solution s^k to the system of linear equations

$$J_F(x^k)\ s^k = -F(x^k)$$

New iterate $x^{k+1} = x^k + s^k$

Excellent local convergence properties

Standard Assumptions

Standard assumptions on the function $F : X \to \mathbb{R}^n$ where $X \subset \mathbb{R}^n$

(1) The system of equations $F(x) = 0$ has a solution x^*.

(2) The function $J_F : X \to \mathbb{R}^{n\times n}$ is Lipschitz continuous with Lipschitz constant γ.

(3) The matrix $J_F(x^*)$ is nonsingular.

Local Convergence

Open neighborhood around a point y

$$\mathcal{B}_\delta(y) = \{x \ : \ ||x - y|| < \delta\}$$

Classical local convergence result for Newton's method

Theorem. Suppose the standard assumptions hold. Then there exists $\delta > 0$ such that for $x^0 \in \mathcal{B}_\delta(x^*)$ the Newton iteration

$$x^{k+1} = x^k - [J_F(x^k)]^{-1} F(x^k)$$

is well-defined (that is, $J_F(x^k)$ is nonsingular) and generates a sequence of iterates $x^0, x^1, \ldots, x^k, x^{k+1}, \ldots$ which converges quadratically to x^*, that is, for all sufficiently large k, there is $K > 0$ such that

$$||x^{k+1} - x^*|| \leq K ||x^k - x^*||^2.$$

Solving Cournot Game ($N = 4$) with Newton's Method

k	q_i^k	$\max_i \lvert q_i^k - q_i^{k-1} \rvert$	$\lVert F(q^k) \rVert$
0	10	–	164.70
1	24.6208	14.6208	4.0967
2	24.9999	0.3791	1.1675 (−3)
3	25.0000	1.0810 (−4)	9.3476 (−11)
4	25.0000	8.6615 (−12)	2.0409 (−14)

Shortcomings of Newton's Method

If initial guess x^0 is far from a solution Newton's method may behave erratically; for example, it may diverge or cycle

If $J_F(x^k)$ is singular the Newton step may not be defined

It may be too expensive to compute the Newton step s^k for large systems of equations

The root x^* may be degenerate ($J_F(x^*)$ is singular) and convergence is very slow

Practical variants of Newton-like methods overcome most of these issues

Merit Function for Newton's Method

General idea: Obtain global convergence by combining the Newton step with line-search or trust-region methods from optimization

Merit function monitors progress towards root of F

Most widely used merit function is sum of squares

$$M(x) = \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^{n} F_i^2(x)$$

Any root x^* of F yields global minimum of M

Local minimizers with $M(x) > 0$ are not roots of F

$$\nabla M(\tilde{x}) = J_F(\tilde{x})^\top F(\tilde{x}) = 0$$

and so $F(\tilde{x}) \neq 0$ implies $J_F(\tilde{x})$ is singular

Line-Search Method

Newton step

$$J_f(x^k)\ s^k = -F(x^k)$$

yields a descent direction of M as long as $F(x^k) \neq 0$

$$\left(s^k\right)^\top \nabla M(x^k) = \left(s^k\right)^\top J_F(x^k)^\top F(x^k) = -\|F(x^k)\|^2 < 0$$

Given step length α^k the new iterate is

$$x^{k+1} = x^k + \alpha^k s^k$$

Step Length

Inexact line search condition (Armijo condition)

$$M(x^k + \alpha s^k) \leq M(x^k) + c\,\alpha\,\left(\nabla M(x^k)\right)^\top s^k$$

for some constant $c \in (0, 1)$

Step length is the largest α satisfying the inequality

For example, try $\alpha = 1, \frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^3}, \ldots$

This approach is not Newton's method for minimization

No computation or storage of Hessian matrix

Inexact Line-Search Newton method

Initial. Choose initial iterate $x^{(0)}$, stopping criteria $\varepsilon > 0$ and $\delta > 0$, and $\gamma \in (0, 1]$ for the Armijo rule.

Step 1 Compute the Jacobian $J_F(x^k)$; compute the Newton direction s^k as the solution to the linear system of equations $J_F(x^k)\ s^k = -F(x^k)$

Step 2 (i) $\alpha = 1$;
(ii) If $M(x^k + \alpha s^k) \leq (1 - \gamma\alpha)M(x^k)$ then $\alpha^k = \alpha$ and $x^{k+1} = x^k + \alpha^k s^k$; otherwise replace α by $\alpha/2$ and repeat (ii);

Step 3 Compute $F(x^{k+1})$; if $||F(x^{k+1})|| < \delta$ and $||x^{k+1} - x^k|| < \epsilon(1 + ||x^k||)$ stop; otherwise increase k by 1 and go to Step 1.

Global Convergence

Assumption. The function F is well defined and the Jacobian J_F is Lipschitz continuous with Lipschitz constant γ in an open neighborhood of the level set $\mathcal{L} = \{x : \|F(x)\| \leq \|F(x^0)\|\}$ for the initial iterate x^0. Moreover, $\|J_F^{-1}\|$ is bounded on $\mathcal{L}$. $\square$

Theorem. Suppose the assumption above holds. If the sequence $\{x^k\}$ generated by the inexact line search Newton method with the Armijo rule remains bounded then it converges to a root x^* of F at which the standard assumptions hold, that is, full steps are taken for k sufficiently large and the rate of convergence is quadratic. $\square$

Equilibrium Equations

Bellman equation for each firm

First-order condition w.r.t. quantity q_i

First-order condition w.r.t. investment e_i

Three equations per firm per state

Total of 6 S^2 equations

Solving Large Games in PATH

Generate 6 equations per state with Mathematica

Write output in GAMS format

Call PATH in GAMS

GAMS Code I

```
V1(m1e,m2e) =e= Q1(m1e,m2e)*(1 - Q1(m1e,m2e)/M -
Q2(m1e,m2e)/M) - ((b1*power(Q1(m1e,m2e),2))/2. +
a1*Q1(m1e,m2e))*theta1(m1e) -
((d1*power(U1(m1e,m2e),2))/2. + c1*U1(m1e,m2e))/(-1 + Nst)
+ (beta*((1 - 2*delta + power(delta,2) +
Q2(m1e,m2e)*(delta*kappa - kappa*power(delta,2) +
alpha*kappa*power(delta,2)*U1(m1e,m2e)) + (alpha*delta -
alpha*power(delta,2))*U2(m1e,m2e) +
Q1(m1e,m2e)*(delta*kappa - kappa*power(delta,2) +
power(delta,2)*power(kappa,2)*Q2(m1e,m2e) +
alpha*kappa*power(delta,2)*U2(m1e,m2e)) +
U1(m1e,m2e)*(alpha*delta - alpha*power(delta,2) +
```

GAMS Code II

```
power(alpha,2)*power(delta,2)*U2(m1e,m2e)))*V1(m1e,m2e) +
(delta - power(delta,2) + kappa*power(delta,2)*Q1(m1e,m2e) +
alpha*power(delta,2)*U1(m1e,m2e))*V1(m1e,m2e - 1) + ((alpha
- 2*alpha*delta + alpha*power(delta,2))*U2(m1e,m2e) +
(delta*power(alpha,2) -
power(alpha,2)*power(delta,2))*U1(m1e,m2e)*U2(m1e,m2e) +
Q2(m1e,m2e)*(kappa - 2*delta*kappa + kappa*power(delta,2) +
(alpha*kappa - alpha*delta*kappa)*U2(m1e,m2e) +
U1(m1e,m2e)*(alpha*delta*kappa - alpha*kappa*power(delta,2)
+ delta*kappa*power(alpha,2)*U2(m1e,m2e))) +
Q1(m1e,m2e)*((alpha*delta*kappa -
```

GAMS Code III

```
alpha*kappa*power(delta,2))*U2(m1e,m2e) +
Q2(m1e,m2e)*(delta*power(kappa,2) -
power(delta,2)*power(kappa,2) +
alpha*delta*power(kappa,2)*U2(m1e,m2e))))*V1(m1e,m2e + 1)
+ (delta - power(delta,2) + kappa*power(delta,2)*Q2(m1e,m2e)
+ alpha*power(delta,2)*U2(m1e,m2e))*V1(m1e - 1,m2e) +
power(delta,2)*V1(m1e - 1,m2e - 1) + ((alpha*delta -
alpha*power(delta,2))*U2(m1e,m2e) +
Q2(m1e,m2e)*(delta*kappa - kappa*power(delta,2) +
alpha*delta*kappa*U2(m1e,m2e)))*V1(m1e - 1,m2e + 1) +
((alpha*delta*kappa -
alpha*kappa*power(delta,2))*Q2(m1e,m2e)*U1(m1e,m2e) +
U1(m1e,m2e)*(alpha - 2*alpha*delta + alpha*power(delta,2) +
(delta*power(alpha,2) -
```

GAMS Code IV

```
power(alpha,2)*power(delta,2))*U2(m1e,m2e)) +
Q1(m1e,m2e)*(kappa -  2*delta*kappa + kappa * power(delta,2)
+ Q2(m1e,m2e) * (delta * power(kappa,2) -
power(delta,2)*power(kappa,2) +
alpha*delta*power(kappa,2)*U1(m1e,m2e)) +
(alpha*delta*kappa - alpha*kappa*power(delta,2))*U2(m1e,m2e)
+  U1(m1e,m2e)*(alpha*kappa - alpha*delta*kappa +
delta*kappa*power(alpha,2)*U2(m1e,m2e))))*V1(m1e + 1,m2e)
+ ((alpha*delta -  alpha*power(delta,2))*U1(m1e,m2e) +
Q1(m1e,m2e)*(delta*kappa -  kappa*power(delta,2) +
alpha*delta*kappa*U1(m1e,m2e)))*V1(m1e + 1,m2e - 1) +
((power(alpha,2) - 2*delta*power(alpha,2) +
power(alpha,2)*power(delta,2))*U1(m1e,m2e)*U2(m1e,m2e) +
```

GAMS Code V

```
Q2(m1e,m2e)*U1(m1e,m2e)*(alpha*kappa - 2*alpha*delta*kappa
+ alpha*kappa*power(delta,2) + (kappa*power(alpha,2) -
delta*kappa*power(alpha,2))*U2(m1e,m2e)) +
Q1(m1e,m2e)*((alpha*kappa - 2*alpha*delta*kappa +
alpha*kappa*power(delta,2))*U2(m1e,m2e) +
(kappa*power(alpha,2) -
delta*kappa*power(alpha,2))*U1(m1e,m2e)*U2(m1e,m2e) +
Q2(m1e,m2e)*(power(kappa,2) - 2*delta*power(kappa,2) +
power(delta,2)*power(kappa,2) + (alpha*power(kappa,2) -
alpha*delta*power(kappa,2))*U2(m1e,m2e) +
U1(m1e,m2e)*(alpha*power(kappa,2) -
alpha*delta*power(kappa,2) +
```

GAMS Code VI

```
power(alpha,2)*power(kappa,2)*U2(m1e,m2e)))))*V1(m1e +
1,m2e + 1)))/((1 + kappa*Q1(m1e,m2e))*(1 +
kappa*Q2(m1e,m2e))*(1 + alpha*U1(m1e,m2e))*(1 +
alpha*U2(m1e,m2e)));
```

And that was just one of 6 equations

Results

S	Var	rows	non-zero	dense(%)	Steps	RT (m:s)
20	2400	2568	31536	0.48	5	0 : 03
50	15000	15408	195816	0.08	5	0 : 19
100	60000	60808	781616	0.02	5	1 : 16
200	240000	241608	3123216	0.01	5	5 : 12

Convergence for $S = 200$

Iteration	Residual
0	1.56(+4)
1	1.06(+1)
2	1.34
3	2.04(−2)
4	1.74(−5)
5	2.97(−11)

Functional Forms

Until now quadratic cost functions yield interior solutions

Production cost $C_i(q) = \frac{1}{2} b_i q^2$

Investment cost $CI_i(e) = \frac{1}{S-1}\left(\frac{1}{2} d_i e^2\right)$

Functional Forms

Until now quadratic cost functions yield interior solutions

Production cost $C_i(q) = \frac{1}{2} b_i q^2$

Investment cost $CI_i(e) = \frac{1}{S-1}\left(\frac{1}{2} d_i e^2\right)$

No longer true for other cost functions, e.g. with $a_i, c_i > 0$,

$$C_i(q) = a_i q + \frac{1}{2} b_i q^2, \quad CI_i(e) = c_i e + \frac{1}{S-1}\left(\frac{1}{2} d_i e^2\right)$$

Boundary solutions possible

Complementarity Problems

First-order conditions remain necessary and sufficient
but become nonlinear complementarity conditions

$$0 \leq u_i \perp -\frac{\partial}{\partial u_i} \left\{ \Pi_i \left(u_i, U_{-i} \left(\theta \right) ; \theta \right) + \beta \mathsf{E}_{\theta'} \left\{ V_i \left(\theta' \right) | u_i, U_{-i} \left(\theta \right) ; \theta \right\} \right\} \geq 0$$

Together with value function equations we obtain a
mixed complementarity problem

Initial results indicate that PATH solves MCPs almost as fast as
nonlinear equations

Enhancements

More firms result in larger problems

Transitions beyond $j-1$, j, $j+1$ lead to less sparse problems

"Realistic" functional forms such as CES demand