March 14, 2019

A New Optimization Approach to Maximum Likelihood Estimation of Structural Models

Kenneth L. Judd
Hoover Institution

Che-Lin Su
Northwestern University
2006

Structural Estimation

- Great interest in estimating models based on economic structure
  - Dynamic programming models
  - Games
  - Dynamic stochastic general equilibrium
- Major computational challenge because estimation involves also solving model
- We show that many computational difficulties can be avoided by using optimization tools

Basic Problem - DP Example

- Individual solves a dynamic programming problem
  - Econometrician observes state (with error) and decisions
  - θ is set of parameters
  - σ is a vector of parameters describing the decision rule
  - Rationality imposes a relationship between θ and σ

$$0 = G(\theta, \sigma)$$

- Standard view: Likelihood function for data X

$$L(\theta; X)$$

- MPEC view: Augmented likelihood function for data X

$$\mathcal{L}(\theta, \sigma; X)$$

  where we explicitly express the role of σ
  - We find θ and σ that maximize augmented likelihood but also satisfy rationality (or, equilibrium) conditions

Nested Fixed Point Algorithm

- Given θ, compute σ - in practice, this means writing a program $\sigma = \Sigma(\theta)$
- Note that

$$L(\theta; X) = \mathcal{L}(\theta, \Sigma(\theta); X)$$

- Solve likelihood

$$\max \mathcal{L}(\theta, \Sigma(\theta); X)$$

  - Problems: Must compute $\Sigma(\theta)$ to high accuracy for each θ examined
  - Current View: Erdem et a. (2004):

Estimating structural models can be computationally difficult. For example, dynamic discrete choice models are commonly estimated using the nested fixed point algorithm (see Rust 1994). ....[S]ome recent research ... proposes computationally simple estimators for structural models ... using a two-step approach. .... [But], there can be a loss of efficiency [and]... stronger assumptions about unobserved state variables may be required.

- Is this true?
  - Are structural models so computationally difficult that it is necessary to turn to statistically inferior methods?
  - Are economists experts on computational feasibility?
- In this paper, we argue that the answer to the first question is an emphatic NO!

Simple Consumer Demand Example

- Data and Model
  - Data on demand, q, and price p, but demand is observed with error ε.
  - True demand is $q - \varepsilon$.
  - Assume a parametric form for utility function $u(c;\beta)$ where β is a vector of parameters.
  - Economic theory implies

$$u_c(c;\beta) = u_c(q - \varepsilon;\beta) = p$$

- Standard Approach (from Econ 712, University of Wisconsin, 1979)
  - Assume, for example, a functional form for utility

$$u(c) = c - \beta c^2.$$

  - Solve for demand function

$$c = (1-p)/(2\beta)$$

  - Hence, i'th data point satisfies

$$q_i = (1-p_i)/(2\beta) + \varepsilon_i$$

    for some ε_i.
  - To estimate β, choose β to minimize the sum of squared errors

$$\sum_{i=1} (q_i - (1-p_i)/(2\beta))^2 .$$

- Limitations
  - Need to solver for demand function, which is hard if not impossible
  - For example, suppose
    $$u(c) = c - \beta\left(c^2 + c^4 + c^6\right)$$
    with first-order condition
    $$1 - \beta\left(2c + 4c^3 + 6c^5\right) = p$$
  - There is no closed-form solution for demand function.
  - What were you taught to do in this case? *Change the model*!
- Proper Procedure
  - Deal with the first-order condition directly since it has all the information you can have.
  - Recognize that all you do is find the errors that minimize their sum of squares but are consistent with structural equations.

- Examples
  - For our consumption demand model, this is the problem

$$\begin{aligned} \min_{\varepsilon_i, \beta} \quad & \sum_{i=1} \varepsilon_i^2 \\ \text{s.t.} \quad & u_c\left(q_i - \varepsilon_i; \beta\right) = p_i \end{aligned}$$

  - In the case of the quadratic utility function, this reduces to

$$\begin{aligned} \min_{c_i, \varepsilon_i, \beta} \quad & \sum_{i=1} \varepsilon_i^2 \\ \text{s.t.} \quad & 1 - 2\beta c_i = p_i \\ & q_i = c_i + \varepsilon_i \end{aligned}$$

  - Degree-six utility function produces problem

$$\begin{aligned} \min_{c_i, \varepsilon_i, \beta} \quad & \sum_{i=1} \varepsilon_i^2 \\ \text{s.t.} \quad & 1 - \beta\left(2c_i + 4c_i^3 + 6c_i^5\right) = p_i \\ & q_i = c_i + \varepsilon_i \end{aligned}$$

- Even when you can solve for demand function, you may not want to.
  - Consider the case

$$
\begin{aligned}
u(c) &= c - \beta_1 c^2 - \beta_2 c^3 - \beta_3 c^4 \\
u'(c) &= 1 - 2\beta_1 c - 3\beta_2 c^2 - 4\beta_3 c^3
\end{aligned}
$$

  - Demand function is

$$
\begin{aligned}
q &= \frac{1}{12\beta_3} W - \frac{1}{4}\frac{8\beta_1\beta_3 - 3\beta_2^2}{\beta_3 W} - \frac{1}{4}\frac{\beta_2}{\beta_3} \\
W &= \sqrt[3]{\left(108\beta_1\beta_2\beta_3 - 216\beta_3^2 p + 216\beta_3^2 - 27\beta_2^3 + 12\sqrt{3}\beta_3 Z\right)} \\
Z &= \sqrt{Z_1 + Z_2} \\
Z_1 &= 32\beta_1^3\beta_3 - 9\beta_1^2\beta_2^2 - 108\beta_1\beta_2\beta_3 p + 108\beta_1\beta_2\beta_3 \\
Z_2 &= 108\beta_3^2 p^2 - 216\beta_3^2 p + 27p\beta_2^3 + 108\beta_3^2 - 27\beta_2^3
\end{aligned}
$$

  - Demand function is far costlier to compute than the first-order conditions.

- The (*bad*) habit of only using models with closed-form solutions is unnecessary.
- Gallant knew this 40 years ago.

Basic Problem - DP Example

- Individual solves a dynamic programming problem
- Econometrician observes state (with error) and decisions
- Augmented likelihood function for data X

$$\mathcal{L}\left(\theta, \sigma; X\right)$$

  where θ is set of parameters and σ is decision rule

- Rationality imposes a relationship between θ and σ

$$0 = G\left(\theta, \sigma\right)$$

- We want to find maximum likelihood θ but impose rationality condition

Su-Judd Approach - Use MPEC ideas

- Suppose that an economic model has parameters θ.
- Suppose that equilibrium and optimality imply that the observable economic variables, x, follow a stochastic process parameterized by a finite vector σ.
- The value of σ will depend on θ through a set of equilibrium conditions

$$0 = G(\theta, \sigma)$$

- Denote the augmented likelihood of a data set, X, by $\mathcal{L}(\theta, \sigma; X)$.
- Therefore, maximum likelihood is the constrained optimization problem

$$\begin{array}{ll} \max_{\sigma,\theta} & \mathcal{L}(\theta, \sigma; X) \\ \text{s.t.} & 0 = G(\theta, \sigma) \end{array}$$

- We do not require that equilibrium be defined as a solution to a fixed-point equation.
- We do not need to specify an algorithm for computing σ given θ; good solver is probably better.
- Gauss-Jacobi or Gauss-Seidel methods are often used in economics even though they are at best linearly convergent, whereas good solvers are at least superlinearly convergent locally (if not much better) and have better global properties than GJ and GS typically do.
- Using a direct optimization approach allows one to take advantage of the best available methods from the numerical analysis

SJ Applied to Zurcher

- SJ Timing for estimating three parameters (as in the Rust).

		cpu	Estimates			major	# obj	constr.
T	N	(secs)	RC	θ_1^c	θ_2^c	iter.	evals	evals
1,000	101	0.14	1.112	0.043	0.0029	66	72	72
1,000	201	0.31	1.140	0.055	0.0015	44	59	59
1,000	501	1.65	1.130	0.050	0.0019	58	68	68
1,000	1001	5.54	1.144	0.056	0.0013	58	94	94
10,000	101	0.24	1.236	0.056	0.0015	59	67	67
10,000	201	0.44	1.257	0.060	0.0010	59	67	67
10,000	501	0.88	1.252	0.058	0.0012	35	45	45
10,000	1001	3.47	1.256	0.060	0.0010	39	52	52

- Rust did a two-stage procedure, estimating transition parameters in first stage. We do full ML

		cpu	Estimates					major	# obj	# constr
data	states	(secs)	RC	θ_1^c	θ_2^c	θ_1^p	θ_2^p	iters	evals	evals
1,000	101	0.50	1.107	0.039	0.0030	0.723	0.262	111	137	137
1,000	201	1.13	1.140	0.055	0.0015	0.364	0.600	109	120	120
1,000	501	3.37	1.129	0.050	0.0019	0.339	0.612	115	127	127
1,000	1001	7.56	1.144	0.056	0.0014	0.360	0.608	84	116	116
10,000	101	0.50	1.236	0.052	0.0016	0.694	0.284	76	91	91
10,000	201	0.86	1.257	0.060	0.0010	0.367	0.593	85	97	97
10,000	501	2.73	1.252	0.058	0.0012	0.349	0.596	83	98	98
10,000	1001	19.12	1.256	0.060	0.0010	0.370	0.586	166	182	182

- Problem is solved very quickly.
- Timing is nearly linear in the number of states for modest grid size.
- The likelihood function, the constraints, and their derivatives are evaluated only 45-182 times in this example.
- In contrast, the Bellman operator in NFXP (the constraints here) is evaluated hundreds of times in NFXP.

Resampling

- Resampling can often be used to generate standard errors
- We did 20 resamplings:
  - Five parameter estimation
  - 1000 data points
  - 1001 grid points in DP

Comparisons with NFXP

- We reduce time spent on solving DP
  - Important when DP is hard to solve
  - Less important as the cost of computing likelihood rises
- Closed-form solutions may hurt
  - Substituting out m variables from n squeezes all nonlinearities into the remaining $n - m$ dimensions.
  - Nonlinear elimination of variables reduces number of unknowns but may increase nonlinearity
  - Actually, it is often easier to solve large optimization problems!
  - In optimization, it is nonlinearity, not dimensionality, that makes a problem difficult.
- SJ is far more flexible and easy to implement.
  - Derivatives of both DP solution and likelihood are easier to compute
  - NFXP has a hard time doing analytic derivatives of DP step; uses finite differences
  - This approach encourages one to experiment with many solvers to find the best one

Comparison with Rust Implementation

- Ease of use
  - Rust used Gauss "because: 1. the GAUSS langauge is a high-level symbolic language which enables a nearly 1:1 translation of mathematical formulae into computer code. Matrix operations of GAUSS replace cumbersome do-loops of FORTRAN. 2. GAUSS has built-in linear algebra routines, no links to Lapack needed"
  - SJ: AMPL is also easy to use. All solvers have access to linear algebra routines. AMPL does not have matrix notation, but its approach to matrices, tensors, and indexed sets is very flexible.
- Optimization Method
  - Rust: Outer iteration uses BHHH for a while then switches to BFGS, where the user chooses the switch point.
  - SJ: Use solvers far superior to these methods.
- Derivatives
  - Rust: "The NFXP software computes the value of and its derivatives numerically in a subroutine. This implies that we can numerically compute and its derivatives for each trial value encountered in the course of the process of maximizing. In order to do this, we need a very efficient and accurate algorithm for computing the fixed point."
  - SJ: Use true analytic derivatives. This is done automatically by AMPL, and is done efficiently using ideas from automatic differentiation.

- Dynamic programming method
  - Rust: "Inner Fixed Point Algorithm. Contraction mapping fixed point (poly)algorithm. The algorithm combines contraction iterations with Newton-Kantorovich iterations to efficiently compute the functional fixed point." In Rust, contraction iterations are linearly convergent; quadratic convergence is achieved only at final stage.
  - SJ: We use Newton-style methods that are globally faster than contraction mapping ideas. This is particularly important if β is close to 1, representing short, but realistic, time periods.

CONSTRAINED OPTIMIZATION APPROACHES TO ESTIMATION OF STRUCTURAL MODELS: COMMENT*

FEDOR ISKHAKOV
UNIVERSITY OF NEW SOUTH WALES

JOHN RUST†
GEORGETOWN UNIVERSITY

BERTEL SCHJERNING
UNIVERSITY OF COPENHAGEN

JUNE 2014

Abstract

We revisit the comparison of mathematical programming with equilibrium constraints (MPEC) and nested fixed point (NFXP) algorithms for estimating structural dynamic models by Judd and Su (JS, 2012). They used an inefficient version, NFXP-SA, that relies on the method of successive approximations to solve the fixed point problem. We re-do their comparison using the more efficient version of NFXP that Rust (1987) used, NFXP-NK, which combines successive approximations and Newton-Kantorovich iterations to solve the fixed point problem. MPEC and NFXP-NK are similar in performance when the fixed point dimension and sample size are relatively small and the discount factor is not too close to 1. However for higher dimensional problems, or problems with large sample sizes, NFXP-NK outperforms MPEC by orders of magnitude. MPEC fails to converge with high probability as the fixed point dimension increases, or as the discount factor approaches 1.

Constrained Optimization Approaches to Estimation of Structural Models: Comment*

Fedor Iskhakov
University of New South Wales

Jinhyuk Lee
Korea University

John Rust†
Georgetown University

Bertel Schjerning
University of Copenhagen

Kyoungwon Seo
Korea Advanced Institute of Science and Technology

August 2015

Abstract

We revisit the comparison of mathematical programming with equilibrium constraints (MPEC) and nested fixed point (NFXP) algorithms for estimating structural dynamic models by Su and Judd (SJ, 2012). Their implementation of the nested fixed point algorithm used successive approximations to solve the inner fixed point problem (NFXP-SA). We re-do their comparison using the more efficient version of NFXP proposed by Rust (1987), which combines successive approximations and Newton-Kantorovich iterations to solve the fixed point problem (NFXP-NK). We show that MPEC and NFXP are similar in speed and numerical performance when the more efficient NFXP-NK variant is used.

KEYWORDS: Structural estimation, dynamic discrete choice, NFXP, MPEC, successive approximations, Newton-Kantorovich algorithm.

---

*We are very saddened by the sudden death of Che-Lin Su, who made numerous important contributions to economics and computational economics, including showing how effective MPEC can be for structural estimation of a variety of models in empirical IO. Che-Lin's combination of quiet leadership, energy and creativity was an inspiration to all of us. His untimely death is a huge loss for the entire economics profession. The main results in this comment were independently obtained and submitted as separate papers by Lee and Seo, and Iskahkov, Rust and Schjerning and have been combined as this jointly authored comment. We are grateful to Harry J. Paarsch, Kyoo-il Kim and Daniel Ackerberg for helpful comments. An early version of this paper was presented by Bertel Schjerning at the ZICE2014 workshop at the University of Zurich. We are grateful to participants of the ZICE2014 workshop for helpful feedback that lead to the current draft of this comment. This paper is part of the IRUC research project financed by the Danish Council for Strategic Research (DSF). Financial support is gratefully acknowledged.

†Corresponding author, email: jr1393@georgetown.edu.

Econometrica, Vol. 84, No. 1 (January, 2016), 365–370

COMMENT ON "CONSTRAINED OPTIMIZATION APPROACHES TO ESTIMATION OF STRUCTURAL MODELS"

BY FEDOR ISKHAKOV, JINHYUK LEE, JOHN RUST, BERTEL SCHJERNING, AND KYOUNGWON SEO[1]

We revisit the comparison of mathematical programming with equilibrium constraints (MPEC) and nested fixed point (NFXP) algorithms for estimating structural dynamic models by Su and Judd (2012). Their implementation of the nested fixed point algorithm used successive approximations to solve the inner fixed point problem (NFXP-SA). We redo their comparison using the more efficient version of NFXP proposed by Rust (1987), which combines successive approximations and Newton–Kantorovich iterations to solve the fixed point problem (NFXP-NK). We show that MPEC and NFXP are similar in speed and numerical performance when the more efficient NFXP-NK variant is used.

KEYWORDS: Structural estimation, dynamic discrete choice, NFXP, MPEC, successive approximations, Newton–Kantorovich algorithm.

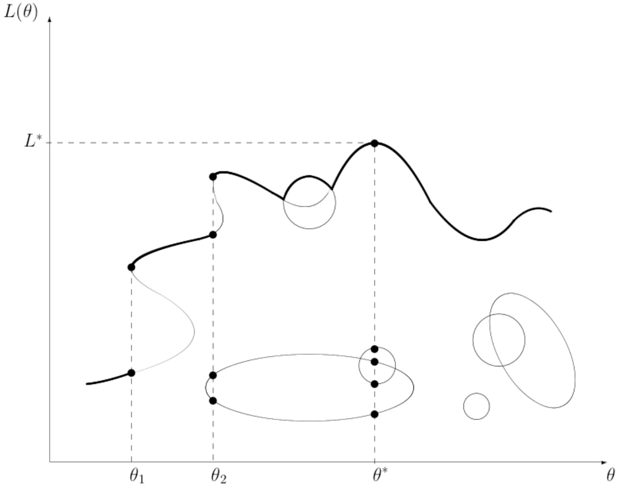
[1]The main results in this comment were independently obtained and submitted as separate papers by Lee and Seo, and Iskahkov, Rust, and Schjerning, and have been combined as this jointly authored comment. We are grateful to Che-Lin Su for substantial input along the course of preparation of this comment. We are very saddened by his untimely death which is a huge loss for the entire economics profession. We are also grateful to Harry J. Paarsch, Kyoo-il Kim, and Daniel Ackerberg for helpful comments. An early version of this paper was presented by Bertel Schjerning at the ZICE2014 workshop at the University of Zurich. We are grateful to participants of the ZICE2014 workshop for helpful feedback that led to the current draft of this comment. This paper is part of the IRUC research project financed by the Danish Council for Strategic Research (DSF). Financial support is gratefully acknowledged.

 DOI: 10.3982/ECTA12605

NFXP Approach to Games

- Follow same procedure:
  - Choose parameter vector each θ
  - Find all the σ that solves $G(\sigma, \theta)$
  - Compute the likelihood of the data for each equilibrium σ, and
  - Report the max.
  - Try more θ.
- Finding all equilibria is an intractable problem!

NFXP Applied to Games with Multiple Equilibria



The SJ Approach to games

- NFXP cannot be used to estimate data from games except for very special cases.
- Suppose that the game has parameters θ representing payoffs, probabilities, and whatever else is not observed directly by the econometrician.
- Let σ denote the equilibrium strategy given θ, and that σ is an equilibrium if and only if

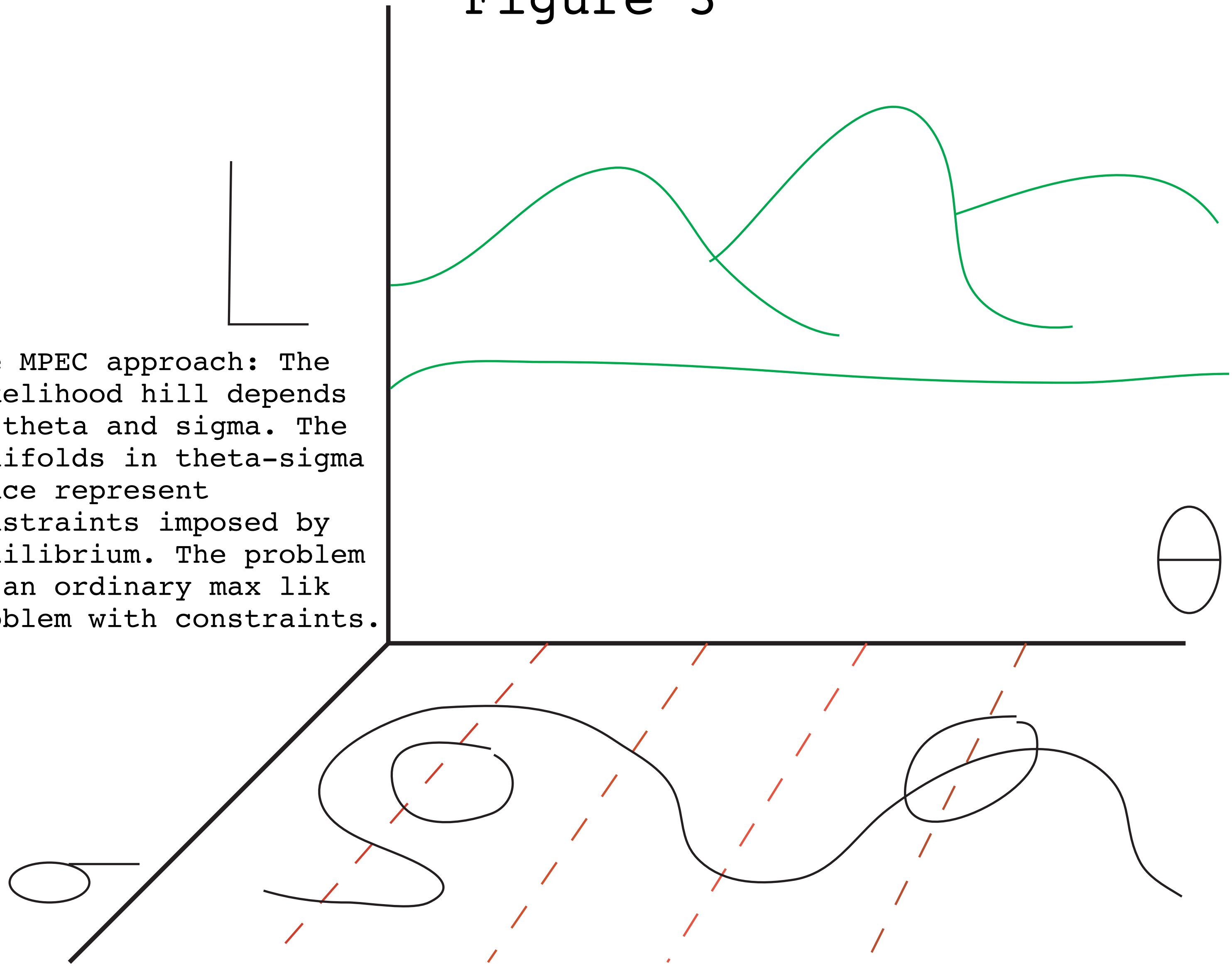
$$0 = G(\sigma, \theta)$$

  for some function G.
- Suppose that likelihood of a data set, x, is $\mathcal{L}(\theta, \sigma, X)$. Therefore, maximum likelihood is the problem

$$\begin{array}{ll} \max\limits_{\sigma,\theta} & \mathcal{L}(\theta, \sigma, X) \\ \text{s.t.} & 0 = G(\sigma, \theta) \end{array}$$

- SJ just sends the problem to good optimization solvers. Multiple equilibria may produce multiple local solutions, but that is a standard problem in maximum likelihood estimation, and would also be a problem for the NFXP approach.

Figure 3



The MPEC approach: The likelihood hill depends on theta and sigma. The manifolds in theta-sigma space represent constraints imposed by equilibrium. The problem is an ordinary max lik problem with constraints.

MPEC

Conclusion

- Structural estimation methods are far easier to construct if one includes the structural equations.
- The numerical algorithm advances of the past forty years (SQP, augmented Lagrangian, interior point, AD, MPCC) makes this tractable
- User-friendly interfaces (e.g., AMPL) makes this as easy to do as Stata, Gauss, and Matlab
- This approach makes structural estimation *really* accessible to a larger set of researchers.