Constrained Optimization Approaches to Estimation of Structural Models

Kenneth L. Judd
Hoover Institution and NBER

Che-Lin Su
Kellogg School of Management
Northwestern University

8th SAET Conference
Kos, Greece

Inverse Optimization Problem on Partially Observed Markov Decision Processes

Kenneth L. Judd
Hoover Institution and NBER

Che-Lin Su
Kellogg School of Management
Northwestern University

8th SAET Conference
Kos, Greece

Structural Estimation

- Great interest in estimating models based on economic structure
  - Dynamic programming models of individual behavior: Rust (1987)
  - Demand estimation: BLP(1995), Nevo(2000)
  - Nash equilibria of games – static, dynamic: AM (2007)?
  - Dynamic stochastic general equilibrium
- General belief: Estimation is a major computational challenge because it involves solving model many times
- Our goal: Reintroduce economists to more efficient ways of estimating structural models (we wish we could say "introduce" but Aitchison and Silvey (1958) and others since beat us)
- Our finding: Many supposed computational "difficulties" can be avoided by using optimization tools developed in numerical analysis over the past 40 years

Standard Problem

- Individual solves an optimization problem
- Econometrician observes states (partially) and decisions
- Current standard approach
  - Structural parameters: θ
  - Behavior (decision rule, strategy, price mapping): σ
  - Equilibrium (optimality or competitive or Nash) imposes relationship between

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

  - Likelihood function for data X and parameters θ

$$L(\theta, \Sigma(\theta); X)$$

  where equilibrium can be represented by a function $\sigma = \Sigma(\theta)$

Nested Fixed Point Algorithm

- Rust (Econometrica, 1987)
  - Given θ, compute $\sigma = \Sigma(\theta)$ – in practice, this means writing a program for $\Sigma(\theta)$
  - Solve likelihood

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

  - Problem with NFXP: Must compute $\Sigma(\theta)$ to high accuracy for each θ examined

Current Views on Structural Estimation

- Erdem et al. (Marketing Letters 2005)

  > 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). This requires solving a dynamic programming problem thousands of times during estimation and numerically minimizing a nonlinear likelihood function....[S]ome recent research ... proposes computationally simple estimators for structural models ... The estimators ... use a two-step approach. ....The two-step estimators can have drawbacks. First, there can be a loss of efficiency. .... Second, stronger assumptions about unobserved state variables may be required. .... However, two-step approaches are computationally light, often require minimal parametric assumptions and are likely to make structural models accessible to a larger set of researchers.

Our Views on Structural Estimation

- This statement says that "Estimating structural models *can* be computationally difficult", particularly if you use NFXP and inefficient numerical methods.
  - More generally, why not find a better alternative instead of giving up on efficient estimation?
- Is this true?
  - (i) Are structural models so computationally difficult that it is necessary to turn to statistically inferior methods?
  - (ii) Do we need " computationally light" approaches to make structural models more "accessible"?

Our Views on Structural Estimation

- This statement says that "Estimating structural models *can* be computationally difficult", particularly if you use NFXP and inefficient numerical methods.
  - More generally, why not find a better alternative instead of giving up on efficient estimation?
- Is this true?
  (i) Are structural models so computationally difficult that it is necessary to turn to statistically inferior methods?
  (ii) Do we need " computationally light" approaches to make structural models more "accessible"?
- Answers:
  (i) No
  (ii) No

MPEC Ideas Applied to Estimation

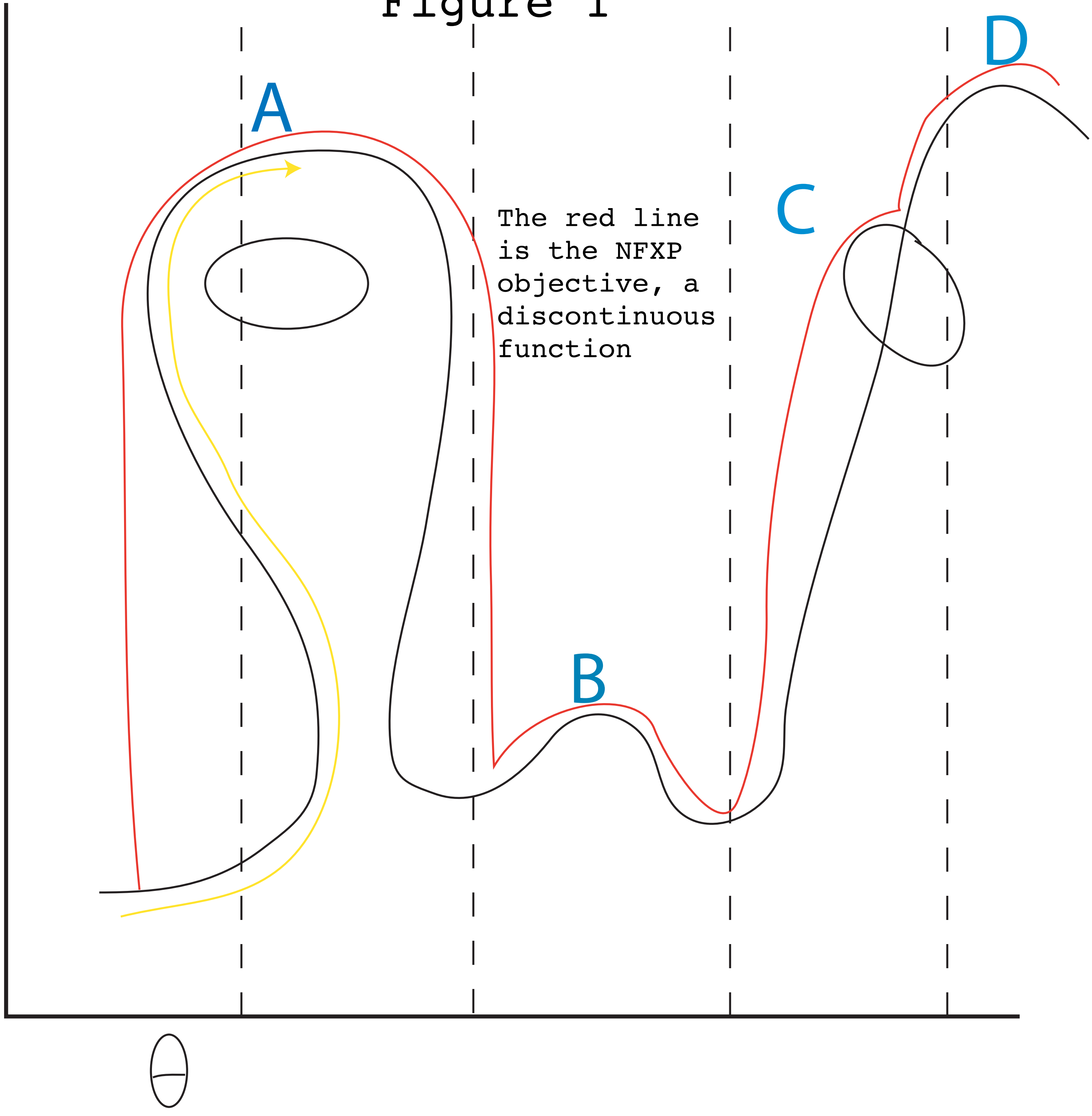
- Suppose that an economic model has parameters θ.
  - Suppose that equilibrium and optimality imply that the observable economic variables, x, follow a stochastic process parameterized by a finite vector σ.
  - The value of σ will depend on θ through a set of equilibrium conditions

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

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

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

Figure 1



A
B
C
D
The red line is the NFXP objective, a discontinuous function
Θ

Figure 2

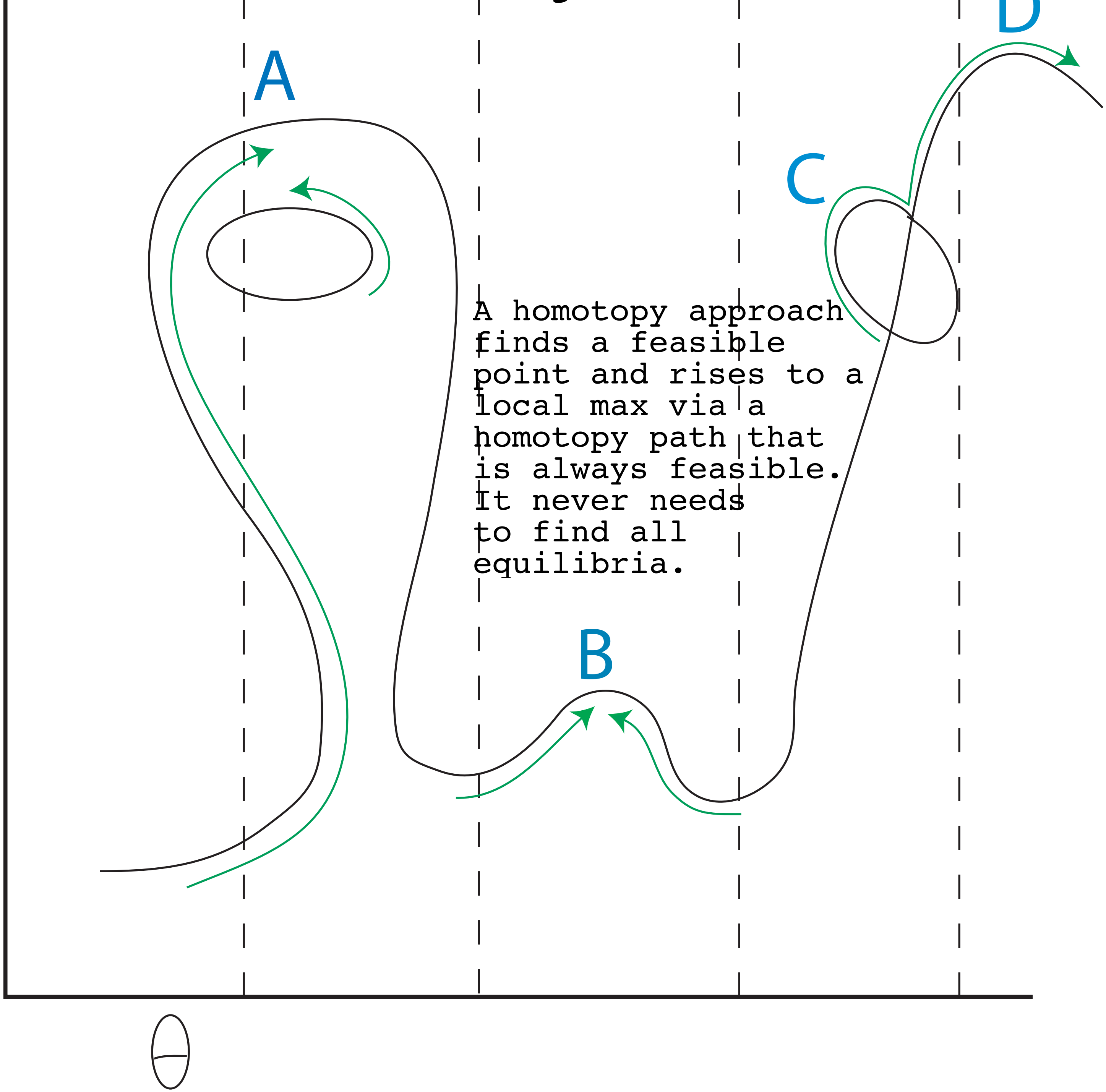
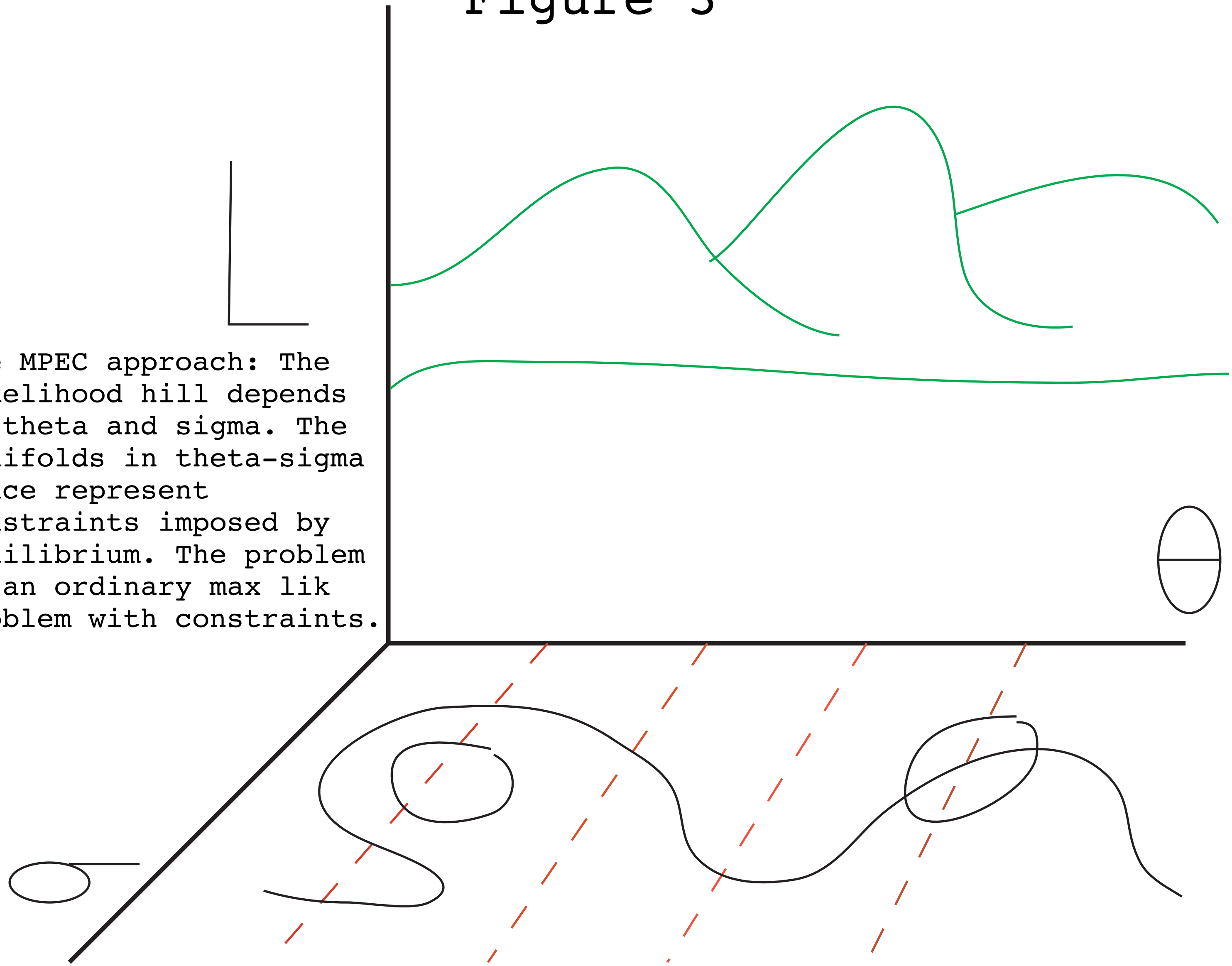

A
B
C
D
A homotopy approach finds a feasible point and rises to a local max via a homotopy path that is always feasible. It never needs to find all equilibria.
Θ

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.

Our Advantages

- 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 will probably do 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

Zurcher's Bus Repair Problem

- Each bus comes in for repair once a month
  - Bus repairman sees mileage since last engine overhaul
  - Repairman chooses between overhaul and ordinary maintenance
  - Repairman has temporary shock to ordinary maintenance cost
  - Repairman solves DP
- Econometrician
  - Observes mileage and decision, but not cost
  - Assumes extreme value distribution
- NFXP
  - Guess θ parameters
  - Solve DP to get decision rule $\sigma = \Sigma(\theta)$ for a given θ
  - Compute likelihood and find best θ

Zurcher Model – Data

Bus #: 5297

events	year	month	odometer at replacement
1st engine replacement	1979	June	242400
2nd engine replacement	1984	August	384900

year	month	odometer reading
1974	Dec	112031
1975	Jan	115223
1975	Feb	118322
1975	Mar	120630
1975	Apr	123918
1975	May	127329
1975	Jun	130100
1975	Jul	133184
1975	Aug	136480
1975	Sep	139429

MPEC Applied to Zucher

- MPEC (Mathematical Program with Equilibrium Constraints)
  - Form 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

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

  - Solve constrained optimization problem

$$\max_{(\theta,\sigma)} \quad \mathcal{L}\left(\theta, \sigma; X\right)$$
$$\text{s.t.} \quad G\left(\theta, \sigma\right) = 0$$

MPEC Applied to Zucher

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

		Estimates			CPU	Major	Evals*	Bell. EQ.
T	N	RC	θ_1^c	θ_2^c	(sec)	Iterations		Error
10^3	101	1.112	0.043	0.0029	0.14	66	72	3.0E-13
10^3	201	1.140	0.055	0.0015	0.31	44	59	2.9E-13
10^3	501	1.130	0.050	0.0019	1.65	58	68	1.4E-12
10^3	1001	1.144	0.056	0.0013	5.54	58	94	2.5E-13
10^4	101	1.236	0.056	0.0015	0.24	59	67	2.9E-13
10^4	201	1.257	0.060	0.0010	0.44	59	67	1.8E-12
10^4	501	1.252	0.058	0.0012	0.88	35	45	2.9E-13
10^4	1001	1.256	0.060	0.0010	1.26	39	52	3.0E-13

*Number of function and constraint evaluations

Five-Parameter Estimates

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

		Estimates					CPU	Maj.	Evals	Bell.
T	N	RC	θ_1^c	θ_2^c	θ_1^p	θ_2^p	(sec)	Iter.		Err.
10^3	101	1.11	0.039	0.0030	0.723	0.262	0.50	111	137	6E−1
10^3	201	1.14	0.055	0.0015	0.364	0.600	1.14	109	120	1E−0
10^3	501	1.13	0.050	0.0019	0.339	0.612	3.39	115	127	3E−1
10^3	1001	1.14	0.056	0.0014	0.360	0.608	7.56	84	116	5E−1
10^4	101	1.24	0.052	0.0016	0.694	0.284	0.50	76	91	5E−1
10^4	201	1.26	0.060	0.0010	0.367	0.053	0.86	85	97	4E−1
10^4	501	1.25	0.058	0.0012	0.349	0.596	2.73	83	98	3E−1
10^4	1001	1.26	0.060	0.0010	0.370	0.586	19.12	166	182	3E−1

Observations

- 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 45-200 times in this example.
- In contrast, the Bellman operator in NFXP (the constraints here) is evaluated hundreds of times in NFXP

Parametric Bootstrap Experiment

- Examine several data sets to determine patterns
- Use the truth in the Rust's example to generate 1 synthetic data set
- Use the estimated values to reproduce 20 independent data sets:
  - Five parameter estimation
  - 1000 data points
  - 201 grid points in DP

ML Parametric Bootstrap Estimates

Table 3: Maximum Likelihood Parametric Bootstrap Results

	Estimates						CPU	Maj.	Evals	Bell
	RC	θ^c_1	θ^c_2	θ^p_1	θ^p_2	θ^p_3	(sec)	Ite		Err.
mean	1.14	0.037	0.004	0.384	0.587	0.029	0.54	90	109	8E−0
S.E.	0.15	0.035	0.004	0.013	0.012	0.005	0.16	24	37	2E−0
Min	0.95	0.000	0.000	0.355	0.571	0.021	0.24	45	59	1E−1
Max	1.46	0.108	0.012	0.403	0.606	0.039	0.88	152	230	6E−0

MPEC Approach to Method of Moments

- Suppose you want to fit moments. E.g., likelihood may not exist in
- Method then is

$$\begin{array}{ll} \min_{(\theta,\sigma)} & \|M(\theta,\sigma) - M(X)\|^2 \\ \text{s.t.} & G(\theta,\sigma) = 0 \end{array}$$

  - Compute moments $M(\theta,\sigma)$ numerically via linear equations in constraints - no simulation
  - Objective function:

$$\begin{aligned} \mathcal{M}(m,M) &= (m_x - M_x)^2 + (m_d - M_d)^2 + (m_{xx} - M_{xx})^2 + (m_{xd} - M_{xd})^2 \\ &+ (m_{dd} - M_{dd})^2 + (m_{xxx} - M_{xxx})^2 + (m_{xxd} - M_{xxd})^2 \\ &+ (m_{xdd} - M_{xdd})^2 + (m_{ddd} - M_{ddd})^2 \end{aligned}$$

Formulation for Method of Moments

- Constraints imposing equilibrium conditions and moment definitions, and computes stationary distribution p

$$\max_{(\theta,\sigma,\Pi,p,m)} \quad \mathcal{M}(m, M)$$

s.t.

$$G(\theta,\sigma) = 0, \quad \Pi = H(\theta,\sigma)$$

$$p^\top \Pi = p^\top, \quad \sum_{x\in Z, d\in\{0,1\}} p_{x,d} = 1$$

$$m_x = \sum_{x,d} p_{x,d}\, x, \; m_d = \sum_{x,d} p_{x,d}\, d$$

$$m_{xx} = \sum_{x,d} p_{x,d}\,(x - m_x)^2, \; m_{xd} = \sum_{x,d} p_{x,d}\,(x - m_x)(d - m_d)$$

$$m_{dd} = \sum_{x,d} p_{x,d}\,(d - m_d)^2$$

$$m_{xxx} = \sum_{x,d} p_{x,d}\,(x - m_x)^3, \; m_{xxd} = \sum_{x,d} p_{x,d}\,(x - m_x)^2(d - m_d)$$

$$m_{xdd} = \sum_{x,d} p_{x,d}\,(x - m_x)(d - m_d)^2, \; m_{ddd} = \sum_{x,d} p_{x,d}\,(d - m_d)^3$$

GMM Parametric Bootstrap Estimates

Table 4: Method of Moments Parametric Bootstrap Results

	Estimates						CPU	Major	Evals	Bell
	RC	θ_1^c	θ_2^c	θ_1^p	θ_2^p	θ_3^p	(sec)	Iter		Err.
mean	1.0	0.05	0.001	0.397	0.603	0.000	22.6	525	1753	7E−0
S.E.	0.3	0.03	0.002	0.040	0.040	0.001	16.9	389	1513	1E−0
Min	0.1	0.00	0.000	0.340	0.511	0.000	5.4	168	389	2E−1
Max	1.5	0.10	0.009	0.489	0.660	0.004	70.1	1823	6851	4E−0

- Solving GMM is not as fast as solving MLE
  - the larger size of the moments problem
  - the nonlinearity introduced by the constraints related to moments, particularly the skewness equations.

MPEC Approach to Games

- Suppose the game has parameters θ.
- Let σ denote the equilibrium strategy given θ; that is, σ is an equilibrium if and only if for some function G

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

- Suppose that likelihood of a data set, X, if parameters are θ and players follow strategy σ is $\mathcal{L}(\theta, \sigma, X)$. Therefore, maximum likelihood is the problem

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

- NFXP requires finding all σ that solve $G(\theta, \sigma) = 0$, compute the likelihood at each such σ, and report the max.

Example: Bertrand Pricing Game

- Bertrand game with 3 types of customers in 4 cities
  - Type 1 customers only want good x

$$Dx_1(p_{x,i}) = A - p_{x,i};\ Dy_1 = 0,\ \text{ for } i = 1, \ldots, 4.$$

  - Type 3 customers only want good y, and have a linear demand curve:

$$Dx_3 = 0;\ Dy_3(p_{y,i}) = A - p_{y,i},\ \text{ for } i = 1, \ldots, 4.$$

  - Type 2 customers want some of both. Let n_i be the number of type 2 customers in a type i city.

$$\begin{aligned} Dx_2(p_{xi}, p_{yi}) &= n_i p_{xi}^{-\sigma} \left(p_{xi}^{1-\sigma} + p_{yi}^{1-\sigma}\right)^{\frac{\gamma-\sigma}{-1+\sigma}} \\ Dy_2(p_{xi}, p_{yi}) &= n_i p_{yi}^{-\sigma} \left(p_{xi}^{1-\sigma} + p_{yi}^{1-\sigma}\right)^{\frac{\gamma-\sigma}{-1+\sigma}} \end{aligned}$$

Example: Bertrand Pricing Game

- Equilibrium possibilities
  - Niche strategy: price high, get low elasticity buyers.
  - Mass market strategy: price low to get type 2 people.
  - Low population implies both do niche
  - Medium population implies one does niche, other does mass market, but both combinations are equilibria.
  - High population implies both go for mass market

Example: Bertrand Pricing Game

- MPEC formulation

$$
\begin{array}{rl}
\min\limits_{(p_{xi},p_{yi},\sigma,\gamma,A,m)} & \sum\limits_{k=1}^{K}\sum\limits_{i=1}^{4}\left((p_{xi}^{k}-p_{xi})^2+(p_{yi}^{k}-p_{yi})^2\right) \\
\text{subject to:} & p_{xi},p_{yi}\geq 0,\ \forall i \\
\text{FOC:} & 0 = MR_y(p_{xi},p_{yi}) = MR_x(p_{xi},p_{yi}),\ \forall\, i \\
\text{global opt :} & (p_{xi}-m)Dx(p_{xi},p_{yi}) \geq (p_j-m)Dx(p_j,p_{yi}),\ \forall\, i,j \\
\text{global opt :} & (p_{yi}-m)Dy(p_{xi},p_{yi}) \geq (p_j-m)Dy(p_{xi},p_j),\ \forall\, i,j
\end{array}
$$

Example: Bertrand Pricing Game

- Data: measurement error in price observations
- Results: no problem getting right estimates.
- Note on Aguirregabiria and Mira (Econometrica, 2007, pp. 1-53):
  - Ag-M say on page 1 that they do not impose a selection criterion
  - Later they reveal that they assume stability under best response
  - Equilibria in our example do not satisfy Ag-M selection criterion

Comparison with Rust Implementation

- Ease of use
  - Rust: Gauss
    - a high-level symbolic language
    - built-in linear algebra routines
  - J-S: AMPL
    - all solvers have access to linear algebra routines
    - flexible approach to matrices, tensors, and indexed sets
- Vectorization
  - Rust: Efficient use of GAUSS requires the user to “vectorize” a program
  - J-S: All vectorization is done automatically in AMPL

Comparison with Rust Implementation

- Optimization Method
  - Rust: BHHH/BFGS
  - J-S: Use solvers far superior to these methods
- Derivatives
  - Rust: compute the value of and its derivatives numerically in a subroutine
  - J-S: Use true analytic derivatives; done automatically and efficiently by AMPL using automatic differentiation.

Comparison with Rust Implementation

- Dynamic programming method
  - Rust: Contraction mapping fixed point (poly)algorithm.
    - combine contraction with Newton-Kantorovich iterations
    - contraction iterations are linearly convergent
    - quadratic convergence is achieved only at final stage.
  - J-S: Newton-style methods
    - globally faster than contraction mapping
    - particularly important if β is close to 1

J-S AMPL Implementation

- Express problem in straightforward language
- Access almost any solver:
  IPOPT, KNITRO, SNOPT, Filter, MINOS, PENNON
- Gradients and Hessians are computed analytically and automatically and efficiently

Future Work and Applications

- Random-Coefficients Demand Estimation (BLP (1995) and Nevo(2000)): J. Fox and C.-L. Su.
- Demand and Supply Joint-Estimation: J.P. Dube, M.A. Vitorino and C.-L. Su
- Estimation of Static Entry Game: Maria Ana Vitoriano, Entry in a Cluster: An Application to the Shopping Center Industry.

Conclusion

- Structural estimation methods are far easier to construct if one uses the structural equations
- The numerical algorithm advances of the past forty years (SQP, augmented Lagrangian, interior point, AD, MPEC) makes this tractable
- Numerical analysis is more useful for empirical economists than new econometric theory
- User-friendly interfaces (e.g., AMPL, GAMS) makes this as easy to do as Stata, Gauss, and Matlab
- This approach makes structural estimation *really* accessible to a larger set of researchers