NAKE Workshop
June 13, 2001
Numerical Methods for Dynamic Economic Analysis
Ken Judd

Projection Methods for Functional Problems

- Many problems require us to solve for some unknown function
  - Dynamic programming
  - Consumption and investment policy functions
  - Pricing functions in asset pricing models
  - Strategies in dynamic games
- The projection method is a robust method for solving such problems

Simple Example: One-Sector Growth

- Optimal growth problem:

$$\max_{c_t} \sum_{t=1}^{\infty} \beta^t u(c_t)$$
$$k_{t+1} = f(k_t) - c_t$$

  - Optimality implies that the optimal consumption path c_t satisfies

$$u'(c_t) = \beta u'(c_{t+1}) f'(k_{t+1})$$

  - Problem: The number of unknowns $c_t,\ t = 1, 2, ...$ is infinite.

Step 0: Express solution in terms of an unknown function

- Consumption function

$$c_t = C(k_t)$$

- $C(k)$ must satisfy the functional equation:

$$\begin{aligned} 0 &= u'(C(k)) - \beta u'(C(f(k) - C(k))) f'(f(k) - C(k)) \\ &\equiv (\mathcal{N}(C))(k) \end{aligned}$$

  - This defines the operator

$$\mathcal{N} : C_+^0 \to C_+^0$$

  - Equilibrium solves the operator equation

$$0 = \mathcal{N}(C)$$

Step 1: Create approximation:

- Find

$$\widehat{C} \equiv \sum_{i=1}^{n} a_i k^i$$

  which "nearly" solves

$$\mathcal{N}\,(\widehat{C}) = 0$$

- Convert an infinite-dimensional problem to a finite-dimensional problem in R^n

  1. No discretization of state space
  2. A form of discretization, but in spectral domain

Step 2: Compute Euler equation error function:

$$\begin{aligned} R\,(k;\ \vec{a}) \;&=\; u'(\widehat{C}(k)) \\ &\quad -\beta u'(\widehat{C}(f(k)-\widehat{C}(k)))f'(f(k)-\widehat{C}(k)) \end{aligned}$$

Step 3: Make $R(\cdot;\vec{a})$ "small" in some sense:

- Least-Squares: minimize sum of squared Euler equation errors

$$\min_{\vec{a}} \int R(\cdot;\vec{a})^2 dk$$

- Galerkin: find $\vec{a}$ to zero weighted averages of Euler equation errors

$$P_i(\vec{a}) \equiv \int R(k;\vec{a})\psi_i(k)dk = 0 \ , \ i = 1, \cdots, n$$

  for n weighting functions $\psi_i(k)$.

- Collocation: find $\vec{a}$ to make Euler equation errors zero at some prespecified k

$$P_i(\vec{a}) \equiv R(k_i;\vec{a}) = 0 \ , \ i = 1, \cdots, n \& \ k_i \in \{k_1, k_2, \cdots, k_n\}$$

Details of $\int ...dk$ computation:

- Exact integration seldom possible in nonlinear problems.
- Use quadrature formulas – they tell us what are *good* points.
- Monte Carlo – often mistakenly used for high–dimension integrals
- Number Theoretic methods – best for large dimension

Details of solving $\vec{a}$:

- Jacobian, $\vec{P}_{\vec{a}}(\vec{a})$, should be well-conditioned.
- Newton's method is quadratically convergent since it uses Jacobian
- Functional iteration (e.g., Deaton, den Haan-Marcet) and time iteration (e.g., Gustafson, Miranda) ignore Jacobian and are linearly convergent.
- Homotopy methods are almost surely globally convergent
- Least squares may be ill-conditioned (that is, be flat in some directions).

Bounded Rationality Accuracy Measure

- Consider the one-period relative Euler equation error:

$$E(k) = 1 - \frac{(u')^{-1}\left(\beta u'\left(C\left(f(k) - C(k)\right)\right) f'\left(f(k) - C(k)\right)\right)}{C(k)}$$

  - Equilibrium requires it to be zero.
  - $E(k)$ is measure of optimization error
    - 1 is unacceptably large
    - Values such as .00001 is a limit for people.
    - $E(k)$ is unit-free.

- Define the L^p, $1 \leq p < \infty$, *bounded rationality accuracy* to be

$$\log_{10} \parallel E(k) \parallel_p$$

- The L^∞ error is the maximum value of $E(k)$ over a range of k.

Numerical Results

- Machine: Compaq 386/20 w/ Weitek 1167 (for JET 1992 paper)
- Speed: Deterministic case: $<$ 15 seconds
- Accuracy: Deterministic case: 8^{th} order polynomial agrees with 250,000–point discretization to within 1/100,000.

Example: Stochastic Dynamic General Equilibrium

Brock-Mirman model

$$\max_{c_t} E\left\{\sum_{t=1}^{\infty} \beta^t u(c_t)\right\}$$
$$k_{t+1} = \theta_t f(k_t) - c_t$$
$$\ln\theta_{t+1} = \rho\ln\theta_t + \epsilon_t$$

- Optimality implies

$$u'(c_t) = \beta E\{u'(c_{t+1})f'(k_{t+1})|\theta_t\}$$

  - Consumption is determined by recursive function

$$c_t = C(k_t, \theta_t)$$

  - $C(k,\theta)$ satisfies functional equation

$$\begin{aligned} 0 = u'(C(k,\theta)) - \beta E\left\{u'\left(C\left(\theta f(k) - C(k,\theta), \theta'\right)\right)\right. \\ \left. \times\ \theta' f'(\theta f(k) - C(k,\theta)) \mid \theta\right\} \end{aligned}$$

  - Transform Euler equation into the more linear form

$$\begin{aligned} 0 &= C(k,\theta) - (u')^{-1}\left(\beta E\left\{u'\left(C(\theta f(k) - C(k,\theta), \tilde\theta)\right)\right.\right. \\ &\qquad \left.\left. \times\ \tilde\theta f'\left(\theta f(k) - C(k,\theta)\right) \mid \theta\right\}\right) \\ &\equiv \mathcal{N}(C)(k,\theta) \end{aligned}$$

- Approximate policy function

$$\widehat{C}(k,\theta;\ \mathbf{a}) \;=\; \sum_{i=1}^{n_k}\sum_{j=1}^{n_\theta} a_{ij}\psi_{ij}(k,\theta)$$

$$\psi_{ij}(k,\theta) \;\equiv\; T_{i-1}\left(2\frac{k-k_m}{k_M-k_m}-1\right)T_{j-1}\left(2\frac{\theta-\theta_m}{\theta_M-\theta_m}-1\right)$$

- Define integrand of expectations

$$\begin{aligned} I(k,\theta,\ \mathbf{a},z) \;=\; & u'\left(\widehat{C}\left(\theta f(k)-\widehat{C}(k,\theta;\ \mathbf{a}), e^{\sigma z}\theta^{\rho},\ \mathbf{a}\right)\right) \\ & \times e^{\sigma z}\theta^{\rho} f'\left(\theta f(k)-\widehat{C}(k,\theta;\ \mathbf{a})\right)\pi^{-\frac{1}{2}} \end{aligned}$$

- $\mathcal{N}\left(\widehat{C}(\cdot,\cdot;\ \mathbf{a})\right)(k,\theta)$ becomes

$$\widehat{C}(k,\theta;\ \mathbf{a})-(u')^{-1}\left(\beta\int_{-\infty}^{\infty} I(k,\theta;\ \mathbf{a},z)\,\frac{e^{-z^2/2}}{\sqrt{2\pi}}\,dz\right)$$

- Use Gauss-Hermite quadrature over z:

$$\int_{-\infty}^{\infty} I(k,\theta,\ \mathbf{a},z)\,\frac{e^{-z^2/2}}{\sqrt{2}}\,dz \doteq \sum_{j=1}^{m_z} I\left(k,\theta,\ \mathbf{a},\sqrt{2}z_j\right)\omega_j$$

  where ω_j, z_j are Gauss-Hermite quadrature weights and points.

- The computable residual function is

$$\begin{aligned} R(k,\theta;\ \mathbf{a}) \;=\; & \widehat{C}(k,\theta;\ \mathbf{a})-(u')^{-1}\left(\beta\sum_{j=1}^{m_z} I\left(k,\theta,\ \mathbf{a},\sqrt{2}z_j\right)w_j\right) \\ \equiv\; & \widehat{\mathcal{N}}\left(\widehat{C}(\cdot,\cdot;\ \mathbf{a})\right)(k,\theta). \end{aligned}$$

- Fitting Criteria:
  - Collocation:
    - Choose n_k capital stocks, $\{k_i\}_{i=1}^{n_k}$, and n_θ productivity levels, $\{\theta_i\}_{j=1}^{n_\theta}$
    - Find **a** such that

$$R(k_i, \theta_j;\ \mathbf{a}) = 0,\ i = 1, \cdots, n_k,\ j = 1, \cdots, n_\theta$$

  - Galerkin:
    - Compute the $n_k n_\theta$ projections

$$P_{ij}(\ \mathbf{a}) \equiv \int_{k_m}^{k_M} \int_{\theta_m}^{\theta_M} R(k, \theta;\ \mathbf{a}) \psi_{ij}(k, \theta) d\theta dk$$

    - Approximate projections by Gauss-Chebyshev quadrature

$$\hat{P}_{ij}(\mathbf{a}) \equiv \sum_{\ell_k=1}^{m_k} \sum_{\ell_\theta=1}^{m_\theta} R(k_i, \theta_j;\ \mathbf{a}) \psi_{ij}(k_{\ell_k}, \theta_{\ell_\theta}),$$

    where

$$\begin{aligned} k_{\ell_\theta} &= k_m + \frac{1}{2}(k_M - k_m)\left(z_{\ell_k}^{m_k} + 1\right),\ \ell_k = 1, \ldots, m_k \\ \theta_{\ell_\theta} &= \theta_m + \frac{1}{2}(\theta_M - \theta_m)\left(z_{\ell_\theta}^{m_\theta} + 1\right),\ \ell_\theta = 1, \ldots, m_\theta \\ z_\ell^n &\equiv \cos\left(\frac{(2i-1)\pi}{2n}\right),\ \ell = 1, \ldots, n \end{aligned}$$

    - Coefficients, **a**, are fixed by the system

$$\hat{P}_{ij}(\mathbf{a}) = 0,\ i = 1, \cdots, n_k,\ j = 1, \cdots, n_\theta$$

- Bounded Rationality Accuracy Measure

  – Consider the computable Euler equation error

  $$E(k,\theta) = \frac{\widehat{\mathcal{N}}(\widehat{C})(k,\theta)}{\widehat{C}(k,\theta)}$$

  where $\widehat{\mathcal{N}}$ uses some integration formula for $E\{\cdot\}$; need not be the same as used in computing $R(k,\theta;\mathbf{a})$. In fact, should use better one.

  – Define the L^p, $1 \leq p < \infty$, *bounded rationality accuracy* to be

  $$\log_{10} \parallel E(k) \parallel_p$$

- Numerical Results

  – Machine: Compaq 386/20 (old, but relative speeds are still valid)

  – Basis: Chebyshev polynomials

  – Initial guess: Linear rule through deterministic steady state and zero.

  – $k \in [.333, 2.000]$

  – Method: Collocation and Galerkin.

  – Newton's method solved projection equations, $P_i(\mathbf{a}) = 0$, for $\mathbf{a}$.

  – Speed: Stochastic case: under two minutes for a 60 parameter fit.

  – Errors: 2% for 6 parameter fit, .1% for 60 parameter fit – about a penny loss per $10,000 dollar expenditure

  – Orth. poly. + orthog. collocation + Gaussian quad. + Newton outperforms naive methods by factor of 10 or greater; exceeded Monte Carlo methods by factor of 100+.

  – $\widehat{C}(k,\theta;\ \mathbf{a})$ is an ϵ-equilibrium with small ϵ – a bounded rationality interpretation.

Table 17.1: $\log_{10}$ Euler Equation Errors

γ	ρ	σ	$\parallel E \parallel_\infty$	$\parallel E \parallel_1$	$\parallel E \parallel_\infty$	$\parallel E \parallel_1$
			(2, 2, 2, 2)*		(4, 3, 4, 3)	
−15.00	0.80	0.01	−2.13	−2.80	−3.00	−3.83
−15.00	0.80	0.04	−1.89	−2.54	−2.44	−2.87
−15.00	0.30	0.04	−2.13	−2.80	−2.97	−3.83
− 0.10	0.80	0.01	−0.01	−1.22	−1.68	−2.65
− 0.10	0.80	0.04	0.01	−1.19	−1.48	−2.22
− 0.10	0.30	0.04	0.18	−1.22	−1.63	−2.65
			(7, 5, 7, 5)		(7, 5, 20, 12)	
−15.00	0.80	0.01	−4.28	−5.19	−4.43	−5.18
−15.00	0.80	0.04	−3.36	−4.00	−3.30	−3.95
−15.00	0.30	0.04	−4.24	−5.19	−4.38	−5.18
− 0.10	0.80	0.01	−3.40	−4.37	−3.47	−4.39
− 0.10	0.80	0.04	−2.50	−3.22	−2.60	−3.17
− 0.10	0.30	0.04	−3.43	−4.37	−3.49	−4.39
			(10, 6, 10, 6)		(10, 6, 25, 15)	
−15.00	0.80	0.01	−5.48	−6.43	−5.61	−6.42
−15.00	0.80	0.04	−3.81	−4.38	−3.88	−4.37
−15.00	0.30	0.04	−5.45	−6.43	−5.57	−6.42
−0.10	0.80	0.01	−5.09	−6.12	−5.17	−6.15
−0.10	0.80	0.04	−2.99	−3.68	−3.09	−3.64
−0.10	0.30	0.04	−5.17	−6.12	−5.23	−6.14

*$(n_k, n_\theta, m_k, m_\theta)$

Table 17.2: Alternative Implementations

$n_k = 7, n_\theta = 5, m_k = 7, m_\theta = 5$

γ	ρ	σ	G[a] error[e]		P[b] error	time	U[c] error	time	UP[d] error	time
−15	.8	.04	−3.18	1:15	−2.13	:40	−3.06	1:05	−2.19	:44
	.3	.01	−4.35	:11	−4.35	:52	−4.07	:08	−4.07	1:47
−.9	.8	.04	−3.43	:05	−3.43	:19	−3.42	:08	−3.42	:39
	.3	.01	−4.03	:07	−4.03	:30	−3.76	:07	−3.76	1:10
−.1	.8	.04	−2.50	:07	−2.50	:41	−2.52	:06	−2.52	:42
	.3	.01	−3.42	:08	−3.42	1:30	−3.18	:07	−3.18	:24

$n_k = 10, n_\theta = 6, m_k = 25, m_\theta = 15$

γ	ρ	σ	G[a] error[e]		P[b] error	time	U[c] error	time	UP[d] error	time
−15	.8	.04	−3.87	4:20	−3.90	24:44	−3.90	3:41	−3.36	42:15
	.3	.01	−5.68	2:19	−5.14	11:31	−5.49	2:14	−5.30	8:06
−.9	.8	.04	−4.00	1:31	−4.00	5:17	−4.01	1:31	−4.01	5:02
	.3	.01	−5.40	1:23	−4.63	7:13	−5.25	1:20	−5.13	6:01
−.1	.8	.04	−3.09	1:31	−3.09	9:16	−3.10	1:32	−3.07	12:01
	.3	.01	−5.27	1:32	−4.02	7:25	−5.09	1:27	−3.27	8:32

[a] Chebyshev polynomial basis, Chebyshev zeroes used in evaluating fit
[b] Ordinary polynomial basis, Chebyshev zeroes used in evaluating fit
[c] Chebyshev polynomial basis, uniform grid points
[d] Ordinary polynomial basis, uniform grid points
[e] error measure is $\| E(k) \|_\infty$

General Projection Method

Step 0: Express solution in terms of unknown functions

$$0 = \mathcal{N}(h)$$

- The $h(x)$ are decision and price rules expressing the dependence on the state x

  1. consumption as a function of wealth
  2. aggregate investment as a function of current state
  3. equilibrium price as a function of.all information
  4. firm investment as a function of his and rivals' current capital stock

- The functions h express

  1. agents on their demand curve
  2. firms on their product supply and factor demand curves
  3. market clearing
  4. value functions from dynamic programming problems

- The collection of conditions $0 = \mathcal{N}(h)$ express equilibrium.

Step 1: Choose space for approximation:

- Basis for approximation for h:

$$\{\varphi_i\}_{i=1}^{\infty} \equiv \Phi$$

- Norm: $\langle \cdot, \cdot \rangle : C_+^0 \times C_+^0 \to R$
- Basis should be
  - complete in space of C_+^0 functions
  - orthogonal w.r.t. $\langle \cdot, \cdot \rangle$ norm
  - easy to compute
  - easily approximate solutions for problem
- Norms are often of form $\langle f, g \rangle = \int_D f(x)g(x)w(x)dx$ for some $w(x) > 0$
- Goal: Find an

$$\widehat{h} \equiv \sum_{i=1}^{n} a_i \, \varphi_i$$

  which "nearly" solves $\mathcal{N}\left(\widehat{h}\right) = 0$
- We have converted an infinite-dimensional problem to a finite-dimensional problem in R^n – discretization in a functional (spectral) sense.

- Example Bases:
  1. $\Phi = \{1, k, k^2, k^3, \cdots\}$
  2. $\Phi = \{\sin\ k,\ \sin\ 2k, \cdots\}$: Fourier – (periodic problems)
  3. $\varphi_n = \cos(n\ \cos^{-1} k)$: Chebyshev polynomials – (for smooth, nonperiodic problems)
  4. Splines (smooth generalizations of step and tent functions)
- Nonlinear generalization
  1. For some parametric form, $\Phi(x; a)$
  $$\widehat{h}(x; a) \equiv \Phi(x; a)$$
  2. Examples:
     - Neural networks
     - Wavelets
     - Rational functions
  3. Goal: Find an
  $$\widehat{h} \equiv \Phi(x; a)$$
  which "nearly" solves $\mathcal{N}\ (\widehat{h}) = 0$
- Promising direction but little known about practicality and performance.

Step 2: Compute residual function:

$$R(\cdot, a) = \widehat{\mathcal{N}}(\widehat{h}) \doteq \mathcal{N}(\widehat{h}) \doteq \mathcal{N}(h)$$

Step 3: Choose $\vec{a}$ so that $R(\cdot;\vec{a})$ is "small" in $\langle \cdot, \cdot \rangle$.

- Alternative Criteria:

  1. Least-Squares

  $$\min_{\vec{a}} \ \langle R(\cdot;\vec{a}),\ R(\cdot;\vec{a})\rangle$$

  2. Galerkin

  $$P_i(\vec{a}) \equiv \langle R(\cdot;\vec{a}),\ \varphi_i\rangle \ = 0, i = 1, \cdots, n$$

  3. Method of Moments

  $$P_i(\vec{a}) \equiv \left\langle R(\cdot;\vec{a}), k^{i-1} \right\rangle = 0 \ , \ i = 1, \cdots, n$$

  4. Collocation

  $$P_i(\vec{a}) \equiv R(k_i;\vec{a}) = 0 \ , \ i = 1, \cdots, n, \ k_i \in \{k_1, k_2, \cdots, k_n\}$$

  5. Orthogonal Collocation (a.k.a. Pseudospectral)

  $$P_i(\vec{a}) \equiv R(k_i;\vec{a}) = 0 \ , \ i = 1, \cdots, n, \ k_i \in \{k : \varphi_n(k) = 0\}$$

- Use good integration methods to solve integrals

- Use good equation-solvers to find $\vec{a}$

Solving Asset Pricing Models with Asymmetric Information

- General Problem
  - Different agents have different information
  - Question: how much information revealed by information
- Grossman (1976)
  - Find all information revealed by trading
  - Finds no incentive to acquire information
  - Assumed special functional forms
  - Assumed and limited type of assets
- More generally
  - Equilibrium is fully revealing if prices are continuous and states finite
  - Equilibrium is often not fully revealing
  - Need more general models which are tractable

- A Gamma-Gaussian Model
  - Investors have W to invest in two assets
  - Safe asset – R
  - Risky asset – $\log \tilde{Z}$ is $N(m, w)$.
  - $m \sim N(\mu_m, \sigma_m^2)$ and $w \sim \Gamma(\alpha, \beta)$ (If the variance of w is zero, we have Grossman's model.)
  - Type i informed traders know $y_i \sim N(m, w)$ plus Gaussian noise, $i = 1, 2, 3$.
  - ω^i is number of shares held by type $i = 1, 2, 3$
  - First-order conditions: Agent i knows p and y_i. His FOC for $\omega^i(p, y_i)$ is

$$\begin{aligned} 0 &= E_{y,Z}\left\{u_i'(C_i(y,Z))(Z - p(y)R) \mid p, y_i\right\}, \; i = 1,2,3 \qquad (27.3.2) \\ C_i(y,Z) &\equiv \left(W - \omega_i(p(y), y_i)p(y)\right)R + \omega_i(p(y), y_i)Z \end{aligned}$$

- Equilibrium:
  - $\omega^i(p, y_i)$ satisfying foc
  - Market-clearing: $p(y)$ satisfying

$$1 = \sum_{i=1,2,3} \omega^i \left(p(y), y_i\right))$$

  for all states y.

Numerical implementation of the conditional expectation:

- Definition of conditional expectation:

$$Z(X) = E\{Y \mid X\}$$

if and only if for all continuous functions ϕ

$$\begin{aligned} 0 &= E\{(Z(X) - Y)\phi(X)\} \\ &= \int (Z(X(\omega)) - Y(\omega))\phi(X(\omega))d\omega \end{aligned}$$

- The definition replaces the conditional expectation with an infinite number of unconditional expectation conditions.

- Numerically: We accept $\widehat{Z}(X)$ as an approximation to $Z(X)$ if

$$\begin{aligned} 0 &= E\left\{(\widehat{Z}(X) - Y)\phi_i(X)\right\} \\ &= \int (\widehat{Z}(X(\omega)) - Y(\omega))\phi_i(X(\omega))d\omega, \; i = 1, ..., n \end{aligned}$$

for a finite number of $\phi_i(\cdot)$ functions.

Numerical Approach

- Parameterize unknown functions
  - $H_i(\cdot)$ denotes the degree i Hermite polynomial
  - price function:

$$p(y_1, y_2, y_3) = \sum_{\substack{0 \le j+k+l \le N_p \\ 0 \le j,k,\ell \le N_p}} a_{jk\ell} H_j(y_1) H_k(y_2) H_\ell(y_3)$$

  - stock demand for a type i investor:

$$\omega_i(p, y_i) = \sum_{0 \le j+k \le N_\theta} b^i_{jk} H_j(p) H_k(y_i),\ i = 1, 2, 3$$

- Goal: determine the $a_{jk\ell}$ and b^i_{jk} coefficients for various N_ω and N_p.
- The first-order-condition for a type i investor
  - Theoretical

$$E_{y,Z} \left\{ U'(\tilde{c}_i)(\tilde{Z} - pR) \mid y_i, p \right\} = 0, \quad i = 1, 2, 3.$$

  - Numerical approximation

$$\begin{aligned} E_{y,Z} \left\{ U'(\tilde{c}_i)(\tilde{Z} - p(y)R) H_j(p(y)) H_k(y_i) \right\} &= 0, \\ j, k \ge 0, j + k &\le N_\theta. \end{aligned} \tag{1}$$

    * The (i, j) condition says that the (excess return)$\times U'(c_i)$ is uncorrelated with $H_j(p(y))H_k(y_i)$.
    * Eq'ns in (1) are integrals over $y_{1,} y_2, y_3, z, m, w$ - six dimensions

- Market clearing
  - Theoretical equilibrium condition: for all states y

$$1=\sum_{i=1,2,3}\omega^{i}\left(p(y),y_{i}\right))$$

  - Numerical approximations are

$$\begin{aligned} E_y\left\{\left(\sum_{i=1}^{3}\omega_i(p(y),y_i)-1\right)H_j(y_1)H_k(y_2)H_\ell(y_3)\right\} &= 0, \\ j,k,l\geq 0, j+k+l &\leq N_p \end{aligned} \quad (2)$$

- Both are approximations
  - First-order conditions are not satisfied in any particular state, just with respect to some averages. Hopefully, our solution will have small f.o.c. errors w.r.t. to other averages.
  - Market-clearing does not hold in each state
    * This has the disadvantage that market clearing will not hold exactly in each state; in fact, it will fail to hold exactly almost always.
    * We hope that the magnitude of the market nonclearing will be very small.

- Numerical Results
  - Four- and Five-digit accuracy on models where we know results
  - Euler equation errors of 1$ per thousand for all of our models
  - Computation time less than 15 minutes on current machines

- Discretization Comments
  - Cannot discretize state space: generic full revelation
  - Must discretize in spectral domain

Extensions

- Endogenous information acquisition
- Other assets - options
- Two-period model with first-period volume information used in second period

Other Applications of Projection Methods

- Multiple agents - Gaspar and Judd
- Dynamic games - Miranda and Rui, Miranda and Vedenov, Doraszelski, Hua and Sibert
- Incomplete asset markets - Judd, Kubler and Schmedders

Summary of Projection Method

- Can be used for any problem with unknown functions
- Uses approximation ideas
- Can exploit a priori information about problem
- Not confined to simple approximations implicit in discretization
- Utilizes standard optimization and nonlinear equation solving software
- Flexible, allowing one to choose from a variety of approximation, integration, and nonlinear equation-solving methods

Projection Method Menu			
Approximation	**Integration**	**Projections**	**Equation Solver**
Piecewise Linear	Newton-Cotes	Galerkin	Newton
Polynomials	Gaussian Rules	Collocation	Powell
Splines	Monte Carlo	M. of Moments	Fixed-pt. iteration
Neural Networks	Quasi-M.C.	Subdomain	Time iteration
Rational Functions	Monomial Rules		Homotopy
Problem Specific	Asymptotics		