Numerical Methods in Economics
MIT Press, 1998

Chapter 17 Notes
Solving Rational Expectations Models

Kenneth L. Judd
Hoover Institution

November 20, 2002

The Lucas Asset Pricing Model

- Model:
  - A single asset paying dividends, y_{t+1}:

$$y_{t+1} = \rho y_t + \varepsilon_{t+1}, \tag{17.1.1}$$

    where ε_t is an i.i.d. innovation process.
  - Representative agent consumes only dividends and has utility function $E\left\{\sum_{t=0}^{\infty} \beta^t u(c_t)\right\}$.
  - p_t price of one unit of asset at time t
  - Supply is normalized to be one.
  - Equilibrium price process must satisfy

$$u'(y_t)p_t = \beta E\left\{u'(y_{t+1})\left(y_{t+1} + p_{t+1}\right) \mid y_t\right\}. \tag{17.1.2}$$

- Recursive solution
  - Let $p(y)$ be the ex-dividend price of a share when the dividend is y
  - Lucas (1978) shows that $p(y)$ solves

$$u'(y_t)p(y_t) = \beta E\left\{u'(y_{t+1})\left(y_{t+1} + p(y_{t+1})\right) \mid y_t\right\}. \tag{17.1.3}$$

  - Rewrite (17.1.3) as

$$p(y_t) = \beta E\left\{\frac{u'(y_{t+1})}{u'(y_t)}\, y_{t+1}\middle| y_t\right\} + \beta E\left\{\frac{u'(y_{t+1})}{u'(y_t)}\, p(y_{t+1})\middle| y_t\right\}. \tag{17.1.4}$$

  - (17.1.4) is a linear Fredholm integral equation

$$p(y) = g(y) + \beta \int K(y,z)\, p(z)\, dz, \tag{17.1.5}$$

    where

$$\begin{aligned} g(y) &= \beta \textstyle\int \frac{u'(z)}{u'(y)}\, z\, q(z|y)\, dz, \\ K(y,z) &= \frac{u'(z)}{u'(y)}\, q(z|y), \end{aligned}$$

    and $q(z|y)$ is the density of the time $t+1$ dividend conditional on the time t dividend.

Solution by Simulation

- Present value arguments show that $p(y)$ is

$$p(y_0) = (u'(y_0))^{-1} E\left\{ \sum_{t=1}^{\infty} \beta^t u'(y_t) \middle| y_0 \right\}. \tag{17.1.6}$$

- Sampling scheme:
  - Draw a sequence of T i.i.d. innovations, ε_t
  - Form dividend sequence $y_t, t = 1, \cdots T$, implied by the process (17.1.1) with y_0 given.
  - Compute the sum

$$P(y_0; \varepsilon) = (u'(y_0))^{-1} \sum_{t=1}^{T} \beta^t u'(y_t). \tag{17.1.7}$$

  - Repeat this for several draws of $\varepsilon \in R^T$; denote the j'th sequence as ε^j, $j = 1, ..., m$.
  - Each ε^j sequence implies a different dividend sequence and a different value for $P(y_0; \varepsilon^j)$.
  - Estimate of $p(y_0)$, denoted $\hat{p}(y_0)$, is the average of the $P(y_0; \varepsilon)$:

$$\hat{p}(y_0) = \frac{1}{m} \sum_{j=1}^{m} P(y_0; \varepsilon^j). \tag{17.1.8}$$

  - Repeat (17.1.8) for N values of y_0, denoted $y_0^i,\ i = 1, ..., N$

- Simulation methods
  - Dominated by other methods if y process has low dimension
  - Are appropriate if desired accuracy is moderate and dividend process is complicated. For example, if

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_\ell y_{t-\ell} + \varepsilon_t,$$

    process is ℓ-dimensional and would challenge other methods.

Lucas Model: Discrete-State Approximation

- Tauchen (1991) recognized that (17.1.3) is a linear Fredholm integral equation of the second kind, and applied the procedures similar to those described in Section 10.8.
- Use Gauss-Hermite quadrature nodes y_i and weights ω_i.
- Prices $p(y_i)$, satisfy the linear equation

$$p(y_i) = g(y_i) + \beta \sum_{j=1}^{n} p(y_j) \frac{u'(y_j)}{u'(y_i)} \omega_j \frac{q(y_j \mid y_i)}{w(y_j)}. \tag{17.1.9}$$

  We solve (17.1.9) to compute approximations to $p(y)$ for $y_i \in Y$; call the solutions $\hat{p}_i$. The procedure so far just approximates $p(y)$ on Y.

- To approximate p globally
  - Use Nystrom extension, equation (10.8.10), to get

$$\hat{p}(y) = g(y) + \beta \sum_{j=1}^{n} \hat{p}_j \frac{u'(y_j)}{u'(y)} \omega_j \frac{q(y_j \mid y)}{w(y_j)}, \tag{17.1.10}$$

    which is defined for all y.
  - This is an example of projection method with Gauss-Hermite polynomials and orthogonal collocation

Example: Stochastic Dynamic General Equilibrium

- Model

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

- Euler equation

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

  – Consumption is determined by recursive function

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

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

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

- Transform Euler equation into the more linear form

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

- 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

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

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

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

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

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

    - Coefficients, $\mathbf{a}$, 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

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

    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

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

- Verify solution: Accept solution to projection equations, $\mathbf{a}$, only if it passes tests
  - Check stability
    - For example, there should be positive savings at low k, high θ
    - Could simulate capital stock process implied by $\widehat{C}(k,\theta;\ \mathbf{a})$ to see if it has a stationary distribution
  - Check Euler equation errors
    - $E(k,\theta)$ should be moderate for most (k,θ) points in $[k_m, k_M] \times [\theta_m, \theta_M]$
    - $E(k,\theta)$ should be small for most (k,θ) points frequently visited
  - If $\widehat{C}(k,\theta;\ \mathbf{a})$ does not pass these tests, go back and use higher values for n_k and n_θ, and increase m_k,and m_θ

- Numerical Results
  - 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 $\mathbf{a}$.
  - Machine: Compaq 386/20 (old, but relative speeds are still valid)
  - 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	$\| E \|_\infty$	$\| E \|_1$	$\| E \|_\infty$	$\| E \|_1$
			$(2,\ 2,\ 2,\ 2)^*$		$(4,\ 3,\ 4,\ 3)$	
-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]		P[b]		U[c]		UP[d]	
			error[e]	time	error	time	error	time	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

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

−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 $\| E(k) \|_\infty$

Table 17.3: Tensor Product vs. Complete Polynomials[a]

			Tensor Product			Complete Polynomials		
γ	ρ	σ	$n=3$	$n=6$	$n=10$	$n=3$	$n=6$	$n=10$
−15.0	.8	.04	−2.34[b]	−3.26	−3.48	−1.89	−3.10	−4.06
			:01[c]	:13	14:21	:03	:07	1:09
−.9	.3	.10	−2.19	−3.60	−5.27	−2.14	−3.55	−5.22
			:01	:08	1:21	:01	:05	:32
−.1	.3	.01	−1.00	−2.84	−5.21	−0.99	−2.83	−5.17
			:01	:08	1:24	:01	:05	:35

[a] Tensor product cases used orthogonal collocation with $n_k = n_\theta = m_k = m_\theta = n$ to identify the n^2 free parameters. Complete polynomial cases used Galerkin projections to identify the $1 + n + n(n+1)/2$ free parameters.

[b] $\log_{10} \parallel E \parallel_\infty$

[c] Computation time expressed in minutes :seconds.

- General Observations:
  - Tensor product of degree n takes more time, but achieves higher accuracy
  - For a specific level of accuracy, complete polynomial method is faster

Examples: Multiagent Dynamic General Equilibrium

- Model:
  - n types of agents, utility functions, $u_i(c)$, $i = 1, 2, ..., n$,
  - Common discount factor β.
  - Equity is the only asset
  - $c_i = C_i(k)$, wealth distribution is $k = (k_1, k_2, ..., k_n)$
- Approximate $c_i = \widehat{C}_i(k, \theta; \mathbf{a})$.
- Euler equation for type $i = 1, 2, ..., n$

$$\begin{aligned} R_i(k, \theta, C) \; &= u_i'(C_i(k, \theta)) - \beta \, E \left\{ u'(C_i(Y(k, \theta) - C(k, \theta), \tilde{\theta})) \right. \\ &\quad \left. \times F_k(Y(k, \theta) - C(k, \theta), \tilde{\theta}) \mid \theta \right\} \end{aligned}$$

where

$$\begin{aligned} Y_i(k, \theta) &= k_i F_1(\overline{k}, \theta) + w(\overline{k}, \theta), \;\; i = 1, .., n \\ w(\overline{k}, \theta) &= F(\overline{k}, \theta) - \overline{k} F_1(\overline{k}, \theta) \\ \overline{k} &\equiv \sum_i k_i. \end{aligned}$$

- Approximate residual function for agents of type $i = 1, 2, ..., n$

$$\begin{aligned}\widehat{R}_i(k, \theta, \widehat{C}(\cdot; \mathbf{a})) &= \widehat{C}_i(k, \theta; \mathbf{a}) - (u_i')^{-1}\left(\beta\, \widehat{E}\left\{u_i'\left(c^+\right) F_k\left(k_\cdot^+, \tilde{\theta}\right) \mid \theta\right\}\right) \\ & \quad c_i^+ \equiv \widehat{C}_i(y^+, \tilde{\theta}; \mathbf{a}) \\ & \quad k^+ \equiv Y(k, \theta; \mathbf{a}) - \widehat{C}(k, \theta; a)\end{aligned}$$

  where $\widehat{E}$ is a numerical approximation of the integral. Use product Gaussian quadrature

- Identifying projections are

$$P_{ij}(\mathbf{a}) \equiv \int_{\theta_m}^{\theta_M} \int_{k_m}^{k_M} \cdots \int_{k_m}^{k_M} \widehat{R}_i(k, \theta, \widehat{C}(\cdot; \mathbf{a}))\, \psi_j(k, \theta)\, w(k, \theta)\, dk_1 \cdots dk_n d\theta$$

  where $i = 1, ..., n$, and $j = 1, ..., m$.

- Let $\widehat{P}(\mathbf{a})$ denote a numerical integration approximation of $P(\mathbf{a})$; we will use product Gaussian quadrature

- Solution chooses $\mathbf{a}$ so that $\widehat{P}(\mathbf{a}) = 0$.

Representation: Tensor vs. Complete Polynomials

- Tensor method:

$$\widehat{C}_i(k,\theta;\mathbf{a}) = \sum_{j_1=0}^{n_k} \cdots \sum_{j_n=0}^{n_k} \sum_{\ell=0}^{n_\theta} a^i_{j_1 \ldots j_n \ell}\, \varphi_{i_1}(k_1) \cdots \varphi_{i_n}(k_n)\, \psi_\ell(\theta), \quad i = 1, ..., n$$

  where $\varphi_i(k_j)$ $(\psi_\ell(\theta))$ is a degree $i-1$ $(\ell-1)$polynomial in k_j (θ) from some orthogonal family.

- Complete polynomial method

$$C_i(k,\theta;\mathbf{a}) = \sum_{\substack{0 \le j_1 + \cdots + j_n + \ell \le d \\ 0 \le j_i, \ell \le d}} a^i_{j_1 \ldots j_n \ell}\, \varphi_{j_1}(k_1) \ldots \varphi_{j_n}(k_n)\, \psi_\ell(\theta)$$

- Number of unknown coefficients are far smaller in complete poly case, but not as flexible.

Solution Methods

- Successive Approximation: at grid of (k, θ) points (e.g., Chebyshev zeroes) and given iteration j for $\mathbf{a}$ (denoted $\mathbf{a}^j$), $\widehat{C}_i(k, \theta; \mathbf{a}^j)$, generate data

$$\begin{aligned} \widehat{C}_i(k, \theta; \mathbf{a}^{j+1}) \;=\; & (u')^{-1}\left(\beta\, \widehat{E}\left\{u'\left(\widehat{C}_i\left(Y(k,\theta) - \widehat{C}_i(k,\theta;\mathbf{a}), \tilde{\theta}; \mathbf{a}\right)\right)\right.\right. \\ & \left.\left. \times F_k\left(Y(k,\theta) - \widehat{C}_i(k,\theta;\mathbf{a}^j), \tilde{\theta}\right) \mid \theta\right\}\right) \end{aligned} \tag{1}$$

  and set coefficients $\mathbf{a}^{j+1}$ through interpolation or regression

- Time Iteration: same procedure except not generate data for $\widehat{C}_i(k, \theta; \mathbf{a}^{j+1})$ by solving

$$\begin{aligned} \widehat{C}_i(k, \theta; \mathbf{a}^{j+1}) \;=\; & (u')^{-1}\left(\beta\, \widehat{E}\left\{u'\left(\widehat{C}_i\left(Y(k,\theta) - \widehat{C}_i(k,\theta;\mathbf{a}^{j+1}), \tilde{\theta}; \mathbf{a}^j\right)\right)\right.\right. \\ & \left.\left. \times F_k\left(Y(k,\theta) - \widehat{C}_i(k,\theta;\mathbf{a}^{j+1}), \tilde{\theta}\right) \mid \theta\right\}\right) \end{aligned} \tag{2}$$

- Newton's Method: just solve nonlinear equations $\widehat{P}(\mathbf{a}) = 0$

Table 5: Time and Accuracy Comparisons

				num.	Newt's Method		Succ.Approx:	
agents	γ	deg	basis	coef's	time	acc'cy	time	accuracy
1	-2	1	t	4	:0.05	-2.7	:0.2	-2.7
			c	3	:0.06	-2.6	:0.4	-2.6
		2	t	9	:0.22	-3.4	:01	-3.4
			c	6	:0.17	-3.3	:01	-3.3
		3	t	16	:0.71	-4.1	:01	-4.1
			c	10	:0.49	-4.0	:02	-4.0
		4	t	25	:02	-4.8	:02	-4.9
			c	30	:0.99	-4.7	:03	-4.6
2	-1	1	t	16	:0.66	-3.1	:01	-3.1
	-2		c	6	:0.38	-2.7	:01	-2.7
		2	t	54	:07	-4.1	:08	-4.1
			c	20	:02	-3.4	:06	-3.4
		3	t	128	1:22	-5.0	:33	-4.5
			c	40	:11	-4.1	:21	-4.1
		4	t	250	12:34	-5.9	1:48	-4.5
			c	70	:45	-4.8	:56	-4.7

Note: "inf" means infeasible. "h hrs $n : m.l$" means "h hours n minutes, $m.l$ seconds".

Table 5: Time and Accuracy Comparisons (Continued)

				num	Newt's Method:		Succ.	Approx.:
agents	γ	deg	basis	coef's	time	accuracy	time	accuracy
3	-1	1	t	48	:07	-3.4	:07	-3.4
	-2		c	15	1.48	-2.8	:05	-2.8
	-3	2	t	243	7:07	-4.6	2:11	-4.5
			c	63	:21	-3.6	:36	-3.6
		3	t	768	inf	inf	19:57	-4.6
			c	105	4:05	-4.3	3:09	-4.3
		4	t	1875	inf	inf	1 hr 56	-4.6
			c	210	46:58	-4.9	12:45	-4.8
4	-.5	1	t	128	1:09	-3.5	:33	-3.5
	-1		c	24	:5.10	-2.9	:13	-2.9
	-2	2	t	972	inf	inf	24:57	-4.6
	-3		c	84	2:47	-3.7	3:04	-3.7
		3	t	4096	inf	inf	7 hr 13	-4.6
			c	224	52:11	-4.4	26:01	-4.4
5	-.5	1	t	320	8:52	-3.6	2:48	-3.6
	-1		c	35	:17.90	-3.0	:38	-3.0
	-2	2	t	3645	inf	inf	5 hr 16	-4.6
	-3		c	140	12:18	-3.8	10:18	-3.8
	-4	3	t	20,480	inf	inf	inf	inf
			c	420	13 hr	-4.5	3 hr 27	-4.5

Note: "inf" means infeasible. "h hrs n : $m.l$" means "h hours n minutes, $m.l$ seconds".

Table 7: Final Comparisons

Method:	**Basis:**	**Solution Method:**	**Advantages:**	**Disadvantages:**
Taylor Series	Complete	Eigenvalues, linear eq'ns	Fast	Local validity
Projection methods	Tensor or complete	Newton	Quadratic conv.	Infeasible for large problems
	Tensor or complete	Successive approx.	Easy Iterations	possible nonconv.

Solving Asymmetric Information Asset Models

- General Problem
  - Different agents have different information
  - Question: how much information is 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, $i = 1, 2, 3$, knows p and y_i. His FOC for $\omega^i(p, y_i)$ is

$$0 = E_{y,Z}\left\{u_i'(C_i(y, Z))(Z - p\,(y)\,R) \mid p, y_i\right\} \qquad (17.3.2)$$
$$C_i(y, Z) \equiv (W - \omega_i(p\,(y)\,, y_i)p\,(y))\,R + \omega_i(p\,(y)\,, y_i)Z$$

- 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

$$E_{y,Z}\left\{U'(\tilde{c}_i)(\tilde{Z} - p(y) R) H_j(p(y)) H_k(y_i)\right\} = 0, \qquad (3)$$
$$j, k \ge 0, j + k \le N_\theta.$$

    * 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 (3) are integrals over $y_{1,} y_2, y_3, z, m, w$ - six dimensions

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

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

  - Numerical approximations are

$$E_y \left\{ \left(\sum_{i=1}^{3} \omega_i (p\left(y\right), y_i) - 1 \right) H_j(y_1) H_k(y_2) H_\ell(y_3) \right\} = 0, \qquad (4)$$
$$j, k, l \geq 0, j + k + l \leq N_p$$

- Both are approximations
  - First-order conditions are satisfied only on average, not in any particular state.
  - Market-clearing does not hold in each state
  - Hope: magnitudes of errors are small and solution is close to true equilibrium.

- 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

Summary of Projection Method

- Can be used for problems with unknown functions
- Uses approximation ideas
- Utilizes standard optimization and nonlinear equation solving software
- Can exploit a priori information about problem
- Flexible: users choose from a variety of approximation, integration, and nonlinear equation-solving methods

Table 17.4: 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		

- Unifies literature: Previous work can be classified and compared

	Choices		
Authors	**Approximation**	**Integration**	**Sol'n Method**
Gustafson(1959)	piecewise linear	Newt.-Cotes	S.A.-time it.
Wright-W.(1982,4)	poly. (of cond. exp.)	Newt.-Cotes	S.A.-time it.
Miranda-H.(1986)	polynomials	Newt.-Cotes	S.A.-learning
Coleman(1990)	finite element	Gaussian	S.A.-time it.
den Haan-M.(1990)	poly. (of cond. exp.)	Sim. M.C.	S.A.-learning
Judd(1992)	orthogonal poly.	Gaussian	Newton