Numerical Methods in Economics
MIT Press, 1998

Chapter 16 Notes
Solution Methods for Perfect Foresight Models

Kenneth L. Judd
Hoover Institution

December 2, 2002

Equilibrium in OLG Models: Time Domain Methods

- Overlapping generations models are commonly used:
  - Agents live for D periods
  - Some bequest motive
- Factor supply
  - Let $K_t^{s,a}\left((R_t, w_t)_{t=1}^{\infty}\right)$ be the supply of capital at time t by agents born in period $t-a+1$. Total supply is

$$K_t^s\left((R_t, w_t)_{t=1}^{\infty}\right) = \sum_{a=1}^{D} K_t^{s,a}\left((R_t, w_t)_{t=1}^{\infty}\right)$$

  - Similarly $L_t^s\left((R_t, w_t)_{t=1}^{\infty}\right)$ represent labor supply at time t.
- Equilibrium: a sequence $(R_t, w_t)_{t=1}^{\infty}$ such that

$$\begin{aligned} R_t &= f_k\left(K_t^s\left((R_t, w_t)_{t=1}^{\infty}\right)\right), t = 1, 2, \cdots, \\ w_t &= f_\ell\left(L_t^s\left((R_t, w_t)_{t=1}^{\infty}\right)\right),\ t = 1, 2, \cdots. \end{aligned} \tag{16.2.2}$$

Algorithm 16.2: OLG Fixed-Point Iteration Algorithm

- Objective: Solve (16.2.2) given the initial capital stock k_0^s.
- Step 0: Choose target horizon T, adjustment factor $\lambda > 0$, and convergence criterion $\varepsilon > 0$. Choose initial guess $(R_t^0, w_t^0)_{t=0}^T$ for the initial factor price path; set R_t and w_t equal to their steady state values for $t > T$.
- Step 1: Given the guess $(R_t^i, w_t^i)_{t=0}^T$, compute the factor supply paths, $(k_t^s, \ell_t^s)_{t=0}^T$, assuming $(R_t, w_t) = (R^*, w^*)$ for $t > T$.
- Step 2: Given factor supply sequence $(k_t^i, \ell_t^i)_{t=0}^T$, compute the factor return sequence $(R_t^+, w_t^+)_{t=0}^T$ implied by the marginal product seqence $(F_k(k_t^i, \ell_t^i), F_\ell(k_t^i, \ell_t^i))_{t=0}^T$.
- Step 3: If each component of $(R_t^+ - R_t^i, w_t^+ - w_t^i)_{t=0}^T$ has magnitude less than ε, STOP. Else, go to Step 4.
- Step 4: Compute a new guess $(R_t^{i+1}, w_t^{i+1})_{t=0}^T$: for each $t = 0, 1, ..., T$,

$$R_t^{i+1} = R_t^i + \lambda(R_t^+ - R_t^i),\ w_t^i + \lambda(w_t^+ - w_t^i)$$

and go to Step 1.

Perfect Foresight Models

- General model

  - $x_t \in R^n$: list of time t values consumption, labor supply, capital stock, output, prices, interest rates, wages, etc.
  - z_t: list of exogenous variables, such as productivity levels, tax rates, monetary growth rates, etc., at time t.
  - Perfect foresight equations

$$
\begin{aligned}
g(t, \overrightarrow{x}, \overrightarrow{z}) &= 0, \ t = 0, 1, 2, ... \\
x_{0,i} &= \overline{x}_{0,i}, \ i = 1, 2, ..., n_I \\
x_t & \quad \text{bounded}
\end{aligned}
$$

- Optimal growth example:

$$
\begin{aligned}
&\max_{c_t} \sum_{t=0}^{\infty} \beta^t u(c_t) \\
&s.t. \quad k_{t+1} = F(k_t) - c_t \\
&\qquad\quad k_0 = \overline{k}_0
\end{aligned}
$$

implies the Euler equation

$$u'(c_t) - \beta u'(c_{t+1}) F'(k_{t+1}) = 0, \ t = 0, 1, 2, ...$$

Eliminate c_t to arrive at equations for k_t

$$
\begin{aligned}
g(t, \overrightarrow{k}) \equiv\ & u'(F(k_t) - k_{t+1}) \\
& -\beta u'(F(k_{t+1}) - k_{t+2}) F'(k_{t+1}) = 0, \ t = 0, 1, ... \qquad (1) \\
k_0 =\ & \overline{k}_0 \\
\lim_{t \to \infty} k_t \to\ & k^{ss}
\end{aligned}
$$

Fair-Taylor Method

- Consider the linear perfect foresight problem

$$y_t = \alpha y_{t+1} + x_t \tag{16.3.1}$$

with stability condition

$$-\infty < \lim_{t\to\infty} y_t < \infty \tag{16.3.2}$$

  - y is the endogenous variable
  - x is a bounded exogenous variable.
  - The true solution is

$$y_0 = \sum_{t=0}^{\infty} \alpha^t x_t.$$

- Fix a horizon T, and terminal value y_{T+1}
  - The system

$$y_t = \alpha y_{t+1} + x_t,\ t = 0, 1, ..., T, \tag{16.3.3}$$

    is $T+1$ equations in the $T+1$ unknowns y_t.
  - Alter T after checks below.

- Make a guess $Y^0_{T,t}$ (say 0) for each y_t, $t = 1, ..., T$;
- $Y^j_{T,t}$ is guess for y_t in iteration j with horizon equal to T.
  - Given $Y^j_{T,t}$, $Y^{j+1}_{T,t}$ is defined by Type I iteration (over t)

$$Y^{j+1}_{T,t} = \alpha Y^j_{T,t+1} + x_t,\ t = 0, 1, ..., T. \tag{16.3.4}$$

  - Repeat (16.3.3) for j, $j = 1, .., T$; these are called *Type II iterations*
  - Type I and II iterations are a Gauss-Jacobi method to solve (16.3.4)

- Adjust the horizon.
  - Let $T+1$ be the terminal period and compute $Y_{T,t}^{T+1}=0$
  - Stop if $Y_{T,t}^{T+1}-Y_{T,t}^{T}$ is small for all t.
- More generally, we solve

$$g(y_t, y_{t+1}, x_t) = 0. \tag{16.3.5}$$

  - Type I and II iterations in Fair-Taylor are implicitly defined by

$$g(Y_{T,t}^{j+1}, Y_{T,t+1}^{j}, x_t) = 0,\ t = 0, .., T,\ j = 0, 1, ... \tag{16.3.6}$$

  - In (16.3.6), each $Y_{T,t}^{j+1}$ is solved by a nonlinear equation solver if a direct solution is not available.
  - Type III iterations increase T until "convergence"
- Reverse shooting method also fixes a horizon T, makes a guess for y_{T+1}, but implements the Gauss-Seidel iteration

$$Y_{T,t} = \alpha Y_{T,t+1} + x_t,\ t = T, T-1, ..., 0, \tag{16.3.7}$$

  - Problem is solved in one pass and with cost proportional to T.
  - In general problem, reverse shooting is the iteration

$$g(Y_{T,t}, Y_{T,t+1}, x_t) = 0,\ t = T, T-1, ..., 0.$$

    and eliminates the need for Type I and II iterations.

Newton Method

- Canonical model

$$g(t, x_t, x_{t+1}) = 0,\ t = 0, 1, 2, ...$$

- Fair-Taylor (Ecm., 1983)
  - A Gauss-Jacobi scheme
  - Slow, possibly nonconvergent
- L-B-J (see Boucekkine, (JEDC, 1995), and Juillard et al (JEDC, 1998))
  - Sparse Jacobian: time t eq'n depends on only (x_t, x_{t+1})

$$J(x) = \begin{pmatrix} g_2(1, x_1, x_2) & g_3(1, x_1, x_2) & 0 & \cdots \\ 0 & g_2(2, x_2, x_3) & g_3(2, x_2, x_3) & \cdots \\ 0 & 0 & g_2(3, x_3, x_4) & \cdots \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \ddots \\ \vdots & \vdots & \vdots & \end{pmatrix}$$

  - Use sparse Newton method from large systems literature

$$\begin{aligned} J(x^k)\Delta &= -g(x^k) \\ x^{k+1} &= x^k + \Delta \end{aligned} \tag{2}$$

  - Faster, more accurate than Fair-Taylor
- Gilli - Paulletto (JEDC, 1998)
  - Solving even sparse linear system is difficult
  - Use Krylov methods to solve for Newton step in Newton scheme

Parametric Path Method

- Illustrate parametric path method with optimal growth problem (1).
- Parameterization
  - Key idea
    - In equilibrium, $\overrightarrow{k}$ evolves smoothly, converges to a steady state
      - Implicitly assumed in Fair-Taylor, L-B-J, A-K, etc.
      - We exploit this implicit assumption
    - Use smooth functional form approximation

$$k_t = \Phi(t; a)$$

    - Find form Φ with coefficient vector $a \in R^m$, $m << T$, such that k_t satisfies (1)
  - Advantages of strategic choice of Φ :
    - A good choice of Φ can greatly reduce number of unknowns
    - A good parameterization can exploit known structure
  - Our implementation
    - λ : the asymptotic rate of convergence to the steady state k^{ss}.
      - λ is often easy to compute
      - rough guess for λ will suffice
    - Initial condition $k(0) = k_0$.
    - Natural parameterization:

$$k(t; a) = \left(k_0 + \sum_{j=1}^{m} a_j t^j \right) e^{-\lambda t} + k^{ss} \left(1 - e^{-\lambda t}\right)$$

imposes both initial and terminal conditions

- Intuitive explanation
  - Define the residual function

$$\begin{aligned}R(t;a)=&u'\left(F(k(t;a))-k(t+1;a)\right)\\&-\beta u'\left(F(k(t+1;a))-k(t+2;a)\right)F'(k(t+1;a))\end{aligned}$$

  - Properties of R
    * By construction, $R(t;a)\rightarrow 0$ as $t\rightarrow\infty$.
    * If $k(t)$ is truth, then $R(t;a)=0$
    * Values of $R(t;a)$ are measures of errors in the equations
  - Solution approach
    * Standard methods aim to find a $k_t\in R^\infty$ such that $\|R(t;a)\|<\varepsilon$ for all t.
    * We aim to find an $a\in R^m$ such that $\|R(t;a)\|<\varepsilon$ for all t if $k(t)$ is given by

$$k(t;a)=\sum_{j=1}^{m}a_j\phi_j(t)+\phi_0(0)^{-1}\left(k_0-\sum_{j=1}^{m}a_j\phi_j(0)\right)+k^{ss}\left(1-e^{-\lambda t}\right)$$

  - Weighted collocation method:
    * Pick some times t_i, $i=1,..,N<<<T$
    * Define loss function

$$Q(a)=\Omega(R(t_1;a),R(t_2;a),..,R(t_N;a))$$

    (which could be $\sum_{i=1}^{N}R(t_i;a)^2$, but there are better choices).
    * Choose a by solving

$$\min_a Q(a)$$

- Finding the Solution
  - Small size suggests Newton (actually, Powell hybrid) can be used
  - Could use homotopy since number of variables is small
- Initial Guesses
  - Simple initial guess
$$k^{init}(t) = k^{ss}$$
  - Natural initial guess:
$$k^{init}(t) = k_0 e^{-\lambda t} + k^{ss}\left(1 - e^{-\lambda t}\right) \tag{3}$$
- Stopping Criterion and Checking the Solution
  - The algorithm checks (1) only at only a small number of t's.
  - Need to check solution at t's we not used.
  - Compute index
$$E = \max_{t=0,1,\ldots,T} \|g(R;a)\|$$
where T is some distant time (T=2500).
  - Traditional methods continue until E is small
  - We increase polynomial degree until E is small

- Numerical Examples
  - Tastes:
  $$u(c) = c^{1+\gamma}/(1+\gamma),\ \gamma = -0.5, -2.0, -5.0.$$
  - Discounting: we choose $\beta = .99$ - a quarterly model.
  - Technology:
  $$F(k) = k + Ak^{\alpha},\ \alpha = .25$$
  where we choose A so that the steady state is $k^{ss} = 1$.
  - $k_0 = .5$.
  - μ - true rate of convergence; easily computed
  - Use 10 different t's
    * Table of t's used in the equations for the solution

γ	λ	t's (which depend solely on λ)
-0.5	μ	1, 6, 16, 32, 55, 87, 131, 194, 293, 513
	$.33\mu$	2, 17, 48, 97, 167, 264, 397, 587, 889, 1555
	3μ	0, 2, 5, 11, 18, 29, 44, 65, 98, 171
-1.1	μ	1, 9, 26, 52, 90, 142, 213, 315, 477, 834
	$.33\mu$	3, 28, 79, 158, 272, 429, 645, 954, 1445, 2528
	3μ	0, 3, 9, 17, 30, 47, 71, 105, 159, 278
-5.0	μ	3, 27, 75, 151, 260, 410, 616, 911, 1380, 2415
	$.33\mu$	9, 81, 228, 458, 788, 1241, 1867, 2762, 4183, 7319
	3μ	1, 9, 25, 50, 87, 137, 205, 304, 460, 805

    * Integration formulas say these t's are "optimal" t's given λ
    * These t's are spread over a wide horizon, but are denser near $t = 0$.

- Errors

  – "Truth" k_t computed via projection methods from Judd (1992).

  – "Truth" had Euler equation errors $<10^{-6}$

  – Error index for approximation $\widehat{k}_t$

$$E = \max_{t=1,..,2500} \frac{\left| k_t - \widehat{k}_t \right|}{k_t}$$

  – Errors: $\lambda = \mu$

Table 1: Maximum errors in $k : \lambda = \mu$

	m :				
γ	1	2	3	4	5
-5.0	1(-3)	6(-4)	3(-4)	2(-4)	1(-4)
-1.1	2(-3)	7(-4)	3(-4)	2(-4)	1(-4)
-0.5	4(-3)	1(-3)	6(-4)	3(-4)	2(-4)

  – Time: Three or fewer iterations

  – Consumption Errors: at most half as large.

- Parametric Path Method for Perfect Foresight Models
  - General Model
    - $x_t \in R^n$: list of time t values consumption, labor supply, capital stock, output, prices, interest rates, wages, etc.
    - z_t: list of exogenous variables, such as productivity levels, tax rates, monetary growth rates, etc., at time t.
    - Perfect foresight equations

$$g(t, \overrightarrow{x}, \overrightarrow{z})=0, \ t = 0, 1, 2, ... \tag{4}$$

$$x_{0,i}=\overline{x}_{0,i}, \ i = 1, 2, ..., n_I \tag{5}$$

$$x_t \quad \text{bounded} \tag{6}$$

  - Key idea
    - Use appropriate functional form approximation

$$x_t = \Phi(t; a)$$

      - Orthogonal polynomials
      - Splines
      - Rational functions
    - Find a such that $x_t = \Phi(t; a)$
      - satisfies initial conditions
      - satisfies asymptotic conditions
      - nearly solves equilibrium equations (4)

Parametric Path Method: Summary

Step 1:	Choose parameterization $x = \Phi(t; a)$.
Step 2:	Form residual function $R(t; a) = g(t, \Phi(.; a), z)$.
Step 3:	Select test functions $p_j(t)$.
Step 4:	Form projections $P_{ij}(a) \doteq \langle R_i(t; a), p_j(t) \rangle$,using numerical integration formulas where necessary.
Step 5:	Solve system of $P_{ij}(a) = 0$ equations plus initial conditions.
Step 6:	Compute $E = \max_{t=0,1,\ldots,T} \|g(t, \overrightarrow{x}, \overrightarrow{z})\|$; accept a if E is sufficiently small; otherwise begin again at Step 1 with a more flexible approximation.

Recursive Models and Dynamic Iteration Methods

- Consider representative agent growth problem

$$\begin{array}{ll} \max_{c_t} \sum_{t=0}^{\infty} \beta^t \, u(c_t), \\ \text{s.t.} \quad k_{t+1} = F(k_t) - c_t. \end{array} \tag{16.4.1}$$

- Equilibrium consumption rule $C(k)$ satisfies

$$u'(C(k)) = \beta u'(C(F(k) - C(k)))F'(F(k) - C(k)) \tag{16.4.2}$$

  – $C(k)$ is zero of operator

$$\begin{aligned} 0 &= u'(C(k)) - \beta u'(C(F(k) - C(k)))F'(F(k) - C(k)) \\ &\equiv (\mathcal{N}(C))(k) \end{aligned} \tag{16.4.3}$$

  – $\mathcal{N}$ is an operator from continuous functions to continuous functions.

- Consider the four occurrences of C and define the operator $\mathcal{F}$:

$$\begin{aligned} 0 &= u'(C_1) - \beta u'(C_2(F - C_3)) \, F'(F - C_4) \\ &\equiv \mathcal{F}(C_1, C_2, C_3, C_4). \end{aligned} \tag{16.4.4}$$

  – We want a function C that solves the equation

$$0 = \mathcal{F}(C, C, C, C) \equiv \mathcal{N}(C). \tag{16.4.5}$$

Time Iteration

- Time iteration implements the iterative scheme

$$0 = u'(C_{i+1}) - \beta u'(C_i(F - C_{i+1}))\, F'(F - C_{i+1}) \tag{16.4.7}$$

  - Intuition: if $C_i(k)$ is tomorrow's consumption policy function, then today's policy, denoted by $C_{i+1}(k)$, must satisfy

$$u'(C_{i+1}(k)) = \beta u'(C_i(F(k) - C_{i+1}(k)))\, F'(F(k) - C_{i+1}(k)). \tag{16.4.9}$$

  - In terms of $\mathcal{F}$, time iteration is the iteration implicitly defined by

$$0 = \mathcal{F}(C_{i+1}, C_i, C_{i+1}, C_{i+1}). \tag{16.4.8}$$

- Convergence
  - Monotonicity property of (16.4.9); that is, if $C_i'(k) > 0$ and $C_i(k) < C_{i-1}(k)$ then $C_{i+1}(k) < C_i(k)$ and C_{i+1} is an increasing function.
  - Monotonicity implies monotone convergence of (16.4.7)
  - However, numerical implementations may introduce numerical error which violates monotonicity.

Fixed-Point Iteration

- Fixed-point iteration applied to (16.4.3) implements the implicit iterative scheme

$$0 = \mathcal{F}(C_{i+1}, C_i, C_i, C_i). \tag{16.4.10}$$

- C_{i+1} is easy to compute since at any k,

$$\begin{aligned} C_{i+1}(k) &= (u')^{-1}\left(\beta u'(C_i(F(k) - C_i(k)))\, F'(F(k) - C_i(k))\right) \\ &\equiv (T_{fp}(C_i))(k) \end{aligned} \tag{16.4.11}$$

- Convergence is not guaranteed

Recursive Models with Nonlinear Equation Methods

- Use nonlinear equations and Chebyshev approximations to solve

$$0 = \mathcal{F}(C, C, C, C). \tag{16.4.10}$$

- No economic "intuition" or "story"; it just works!
- Approximate C with the linear representation

$$\widehat{C}(k\,;\,a) = \sum_{i=1}^{n} a_i \psi_i(k), \tag{16.5.1}$$

  - $\psi_i(k) \equiv T_{i-1}\left(2\frac{k-k_m}{k_M-k_m} - 1\right)$ and n is the number of terms used.
  - Domain D is $[k_m, k_M]$.

- Residual function

$$R(k\,;a) = u'(\widehat{C}(k\,;a)) - \beta u'(\widehat{C}(F(k) - \widehat{C}(k\,;a);a))F'(F(k) - \widehat{C}(k\,;a)). \tag{16.5.2}$$

- Orthogonal collocation chooses k_j and solves

$$R(k_j\,;a) = 0,\ \ j = 1, ..., n. \tag{16.5.3}$$

- Multiple solutions
  - Multiple solutions to first-order conditions exist
  - Only one satisfies global stability
  - If initial guess is close then one typically converges to correct answer
  - Can sometimes avoid bad ones
    - ∗ Specify steady state
    - ∗ Pick functional form which cannot go bad
    - ∗ Alter problem to penalize divergent paths

Coefficients of Solution

- Theoretical predictions
  - Approximation theory says that the Chebyshev coefficients should fall rapidly if $C(k)$ is smooth.
  - Orthogonal basis should imply that coefficients do not change as we increase n.
- Table 16.1 verifies these predictions.

Table 16.1: Chebyshev Coefficients for Consumption Function

k	$n = 2$	$n = 5$	$n = 9$	$n = 15$
1	0.0589755899	0.0600095844	0.0600137797	0.0600137922
2	0.0281934398	0.0284278730	0.0284329464	0.0284329804
3		−0.0114191783	−0.0113529374	−0.0113529464
4		0.0007725731	0.0006990930	0.0006988353
5		−0.0001616767	−0.0001633928	−0.0001634209
6			0.0000427201	0.0000430853
7			−0.0000123570	−0.0000122160
8			0.0000042498	0.0000036367
9			−0.0000011464	−0.0000011212
10				0.0000003557
11				−0.0000001147
12				0.0000000370
13				−0.0000000129
14				0.0000000052
15				−0.0000000015

Each entry is the coefficient of the k'th Chebyshev polynomial (over the interval [.333, 1.667]) in the n-term approximation of the consumption policy function in (4.3) for the case discussed in Section 4.2.

Errors in Consumption Policy Function

- "Truth" computed by a 1,000,000 state discrete approximation
- "True solution" also has some error because of discretization
- Table 16.2 displays difference between approximations and "truth"

Table 16.2: Policy Function Errors

k	y	c	$n=20$	$n=10$	$n=7$	$n=4$	$n=2$
0.5	0.1253211	0.1010611	1(-7)	5(-7)	5(-7)	2(-7)	5(-5)
0.6	0.1331736	0.1132936	2(-6)	1(-7)	1(-7)	2(-6)	8(-5)
0.7	0.1401954	0.1250054	2(-6)	3(-7)	3(-7)	1(-6)	2(-4)
0.8	0.1465765	0.1362965	1(-6)	4(-7)	4(-7)	4(-6)	2(-4)
0.9	0.1524457	0.1472357	1(-6)	3(-7)	3(-7)	5(-6)	2(-4)
1.0	0.1578947	0.1578947	4(-6)	0(-7)	1(-7)	2(-6)	1(-4)
1.1	0.1629916	0.1683016	4(-6)	2(-7)	2(-7)	1(-6)	9(-5)
1.2	0.1677882	0.1784982	3(-6)	2(-7)	2(-7)	4(-6)	7(-6)
1.3	0.1723252	0.1884952	7(-7)	4(-7)	4(-7)	3(-6)	9(-5)