Projection

Wouter J. Den Haan

University of Amsterdam

January 28, 2010

Model

$$
\begin{aligned}
c_t^{-\nu} &= \mathrm{E}_t\left[\beta c_{t+1}^{-\nu}\alpha z_{t+1}k_{t+1}^{\alpha-1}\right] \\
c_t + k_{t+1} &= z_t k_t^{\alpha} \\
\ln(z_{t+1}) &= \rho \ln\left(z_t\right) + \varepsilon_{t+1} \\
& \varepsilon_{t+1} \sim N(0,\sigma^2) \\
& k_1, z_1 \text{ given}
\end{aligned}
$$

Projection Methods

True rational expectations solution:

$$c_t = c(k_t, z_t)$$

- Why a difficult problem to find $c(k_t, z_t)$?

Approximation:

$$c_t = c(k_t, z_t) \approx P_n(k_t, z_t; \eta_n)$$

- $P_n(\cdot)$ from a class of approximating functions (say polynomials or splines) that is dense in a large class of functions (say continuous functions)
- all structural parameter values $(\alpha, \beta, \rho, \sigma)$ take on a fixed numerical value
- n is fixed $\Longrightarrow$ problem is now to solve for η_n, a *finite-dimensional* object

Which equations to use?

What equations to use to solve for N_n elements of η_n?
FOCs at M grid points $\{k_i, z_i\}$ with $M \geq N_n$

$$P_n(k_i, z_i; \eta_n)^{-\nu} =$$

$$E\left[\begin{array}{c} \alpha\beta\times \\ P_n(\{\mathbf{k}'\}_{\mathrm{A}}, \{\mathbf{z}'\}_{\mathrm{B}}; \eta_n)^{-\nu}\times \\ \{\mathbf{z}'\}_{\mathrm{C}}\times \\ (\{\mathbf{k}'\}_{\mathrm{D}})^{\alpha-1} \end{array}\right]$$

- **Goal:** Get for each grid point 1 equation in η_n
- Note that k_i and z_i are known

Which equations to use?

$$P_n(k_i, z_i; \eta_n)^{-\nu} =$$

$$\mathrm{E}\left[\begin{array}{c} \alpha\beta\times \\ P_n(\{z_i k_i^\alpha - P_n(k_i, z_i; \eta_n)\}_{\mathrm{A}}, \{\exp\{\rho \ln(z_i) + \varepsilon'\}\}_{\mathrm{B}}; \eta_n)^{-\nu}\times \\ \{\exp\{\rho \ln(z_i) + \varepsilon'\}\}_{\mathrm{C}} \times \\ \left(\{z_i k_i^\alpha - P_n(k_i, z_i; \eta_n)\}_{\mathrm{D}}\right)^{\alpha-1} \end{array}\right]$$

How to deal with expectations operator?

Let $\{\omega_j, \zeta_j\}_{j=1}^{J}$ be the Hermite Gaussian quadrature nodes

$$P_n(k_i, z_i; \eta_n)^{-\nu} =$$

$$\sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ P_n(z_i k_i^{\alpha} - P_n(k_i, z_i; \eta_n), \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\}; \eta_n)^{-\nu} \times \\ \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\} \times \\ (z_i k_i^{\alpha} - P_n(k_i, z_i; \eta_n))^{\alpha-1} \\ \omega_j / \sqrt{\pi} \end{array} \right]$$

Define error terms

$$e(k_i, z_i; \eta_n) = P_n(k_i, z_i; \eta_n)^{-\nu} -$$

$$\sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ P_n(z_i k_i^{\alpha} - P_n(k_i, z_i; \eta_n), \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\}; \eta_n)^{-\nu} \times \\ \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\} \times \\ (z_i k_i^{\alpha} - P_n(k_i, z_i; \eta_n))^{\alpha-1} \\ \omega_j / \sqrt{\pi} \end{array} \right]$$

How to find coefficients of approximating function?

- True rational expectations solution would give a zero error term $\forall (k_i, z_i)$
- Thus, choose η_n such that error terms are as small as possible.
- **Collacation** $(M = N_n)$: Use equation solver to get errors exactly equal to zero on grid
- **Galerkin** $(M > N_n)$: Use minimization routine (and possibly smart weighting of error terms)
- Iteration procedure $(M \geq N_n)$

Iteration procedure: Construct Grid

- Construct a grid with nodes for k and z
- At the nodes construct the basis functions of $P_n(k, z; \eta_n)$. For example, if

$$P_n(k, z; \eta_n) = \eta_{0,n} + \eta_{k,n}k + \eta_{z,n}z + \eta_{kk}k^2 + \eta_{kz}kz + \eta_{zz}z^2$$

then construct the matrix (where subscripts denote grid numbers)

$$X = \begin{bmatrix} 1 & k_1 & z_1 & k_1^2 & k_1 z_1 & z_1^2 \\ 1 & k_2 & z_2 & k_2^2 & k_2 z_2 & z_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & k_M & z_M & k_M^2 & k_M z_M & z_M^2 \end{bmatrix}$$

and calculate $(X'X)^{-1} X'$

Iteration procedure: Construct Grid

- **Chebyshev nodes:** Using Chebyshev nodes is important. This ensures uniform convergence. With equidistant nodes, e.g., it is possible that the oscilattions between grid point explode as the order of the polynomial increases.
- **Chebyshev polynomials**: If you have (i) no problems finding initial conditions and (ii) only low-order appoximations so that calculating the inverse of $X'X$ can be done accurately, then you can use regular polynomials. Orthogonal Chebyshev polynomials can overcome these problems. They ensure that $X'X$ is diagonal (and trivial to invert). This does require scaling of the state variables so they are between -1 and 1.

Iteration procedure I: Fixed-point Iteration

The value of η_n used in the q^{th} iteration is referred to as η_n^q. Follow the the following iteration scheme until convergence

- At each grid point:
  - Calculate the RHS of the Euler equation using the latest value for η_n, i.e., η_n^{q-1}
  - Use the RHS to calculate the value for c at the grid point
- Use these values of c to obtain an estimate for η_n, $\hat{\eta}_n^q$
  - In case of a polynomial run a regression
  - In case of of a spline the values of c at the nodes are the new values of η_n
- Let $\eta_n^q = \lambda\hat{\eta}_n^q + (1-\lambda)\eta_n^{q-1}$

Iteration procedure I: Fixed-point Iteration

- **Step 1: Calculate current consumption values implied by η_n^{j-1} at each grid point**
  - Use η_n^{q-1} to calculate $k' = z_i k_i^\alpha - P_n(k_i, z_i; \eta_n^{q-1})$
  - Use η_n^{q-1} to calculate $c' = P_n(k', z'; \eta_n^{q-1})$
  - Then, get c_i from

$$(c_i)^{-\nu} = \sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ P_n(z_i k_i^\alpha - P_n(k_i, z_i; \eta_n^{q-1}), \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\}; \eta_n^{q-1})^{-\nu} \times \\ \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\}\times \\ \left(z_i k_i^\alpha - P_n(k_i, z_i; \eta_n^{q-1})\right)^{\alpha-1} \\ \omega_j / \sqrt{\pi} \end{array} \right.$$

Iteration procedure I: Fixed-point iteration

Step 2: Get new estimate for η_n by running a projection step

- Let $Y = [c_1, c_2, \cdots, c_M]'$
- If

$$P_n(k, z; \eta_n) = \eta_{0,n} + \eta_{k,n} k + \eta_{z,n} z + \eta_{kk} k^2 + \eta_{kz} kz + \eta_{zz} z^2$$

  then

$$\hat{\eta}_n^q = (X'X)^{-1} X'Y$$

- If

$$P_n(k, z; \eta_n) = \exp\left(\eta_{0,n} + \eta_{k,n} k + \eta_{z,n} z + \eta_{kk} k^2 + \eta_{kz} kz + \eta_{zz} z^2\right)$$

  then

$$\widehat{\eta}_n^q = (X'X)^{-1} X' \ln(Y)$$

- Note I am allowed to take transformations of RHS & LHS of projection equation because there is not a stochastic error term

Iteration procedure I: Fixed-point iteration

Step 3: Update η_n

$$\eta_n^q = \lambda \widehat{\eta}_n^q + (1-\lambda)\eta_n^{q-1} \quad \text{for } 0 < \lambda \leq 1$$

- Fixed-point iteration does not always converge
  - Choosing a lower value of λ helps but will slow down the convergence
- Alternative is **time iteration**

Iteration procedure I: Time Iteration

Follow the the following iteration scheme until convergence

- At each grid point:
  - In the Euler equation use η_n^{q-1} whenever evaluating *next period's* choices
  - This turns the Euler equation into an equation with only the value of c at this grid point as unknown
  - Use non-linear equation solver to solve for c
- Use these values of c to obtain an estimate for η_n, $\hat{\eta}_n^q$
  - In case of a polynomial run a regression
  - In case of of a spline the values of c at the nodes are the new values of η_n
- Let $\eta_n^q = \hat{\eta}_n^q$ (no dampening necessary)

Fixed point versus time iteration

- Fixed point iteration uses η_n^{q-1} for all terms on the RHS, i.e., both next period's consumption choice and today's capital choice
- Time iteration uses η_n^{q-1} only to evaluate next period's consumption
- The structure of time iteration mimics the choice of value function iteration:
  - next period's behavior described by previous solution for value function
  - Bellman equation used to solve for choice of c and k *simultaneously*

Iteration procedure II: Time Iteration

- Natural interpretation for η_n^{i-1} and η_n^i, namely
  - η_n^{i-1} is tomorrow's policy function and
  - η_n^i is today's policy function
- Time iteration is reliable and convergent (the proof is related to the convergence of value function iteration, which uses the same idea)

Iteration procedure II: Time Iteration

- **Step 1: At each grid point solve** c_i from
  - Then, get c_i from
-

$$(c_i)^{-\nu} = \sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ P_n(z_i k_i^{\alpha} - c_i, \exp\{\rho\ln(z_i) + \sqrt{2}\sigma\zeta_j\}; \eta_n^{i-1})^{-\nu} \times \\ \exp\{\rho\ln(z_i) + \sqrt{2}\sigma\zeta_j\} \times \\ (z_i k_i^{\alpha} - c_i)^{\alpha-1} \\ \omega_j / \sqrt{\pi} \end{array} \right]$$

- Note that c_i shows up on both sides: solving for c_i has become more difficult
- step 2 & 3 are the same, although with time iteration you should not have to use a value for λ less than 1.

Endogenous grid points

- Simple idea: construct grid for k' instead of a grid for k
- Instead of solving for the choice k' given k, we now solve for the value of k that would have led to the choice k'
- In both cases you end up at each grid point with a set of values for k and a set of corresponding values for k'.
- Terminology is a bit confusing: the grid itself is exogenous and fixed but it is for an endogenous variable
- You can use endogenous grid points both with fixed-point and with time iteration (for problem here they turn out to be the same thing)
- The added value with time iteration lies in getting rid of the non-linear problem of solving for today's choices

Endogenous grid points and time iteration

- Time iteration $\Longrightarrow$ Use η_n^{i-1} for tomorrow's choices and η_n^i only for today's choices (which show up on both sides of the policy function
- Then, get c_i from

$$(c_i)^{-\nu} = \sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ P_n(k_i', \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\}; \eta_n^{i-1})^{-\nu}\times \\ \exp\{\rho \ln(z_i) + \sqrt{2}\sigma\zeta_j\}\times \\ (k_i')^{\alpha-1} \\ \omega_j / \sqrt{\pi} \end{array} \right]$$

and k_i from

$$k_i' + c_i = z_i k^{\alpha}$$

Perturbation versus projection

- Nondifferentiabilities

Perturbation versus projection

- Nondifferentiabilities
- Number of state variables

Perturbation versus projection

- Nondifferentiabilities
- Number of state variables
- Constructing the grid can be difficult

Perturbation versus projection

- Nondifferentiabilities
- Number of state variables
- Constructing the grid can be difficult
- Do (lower-order) derivatives at one point capture behavior away from that point?

How do you know you have chosen the right procedure?

- Proposed numerical solution for consumption is equal to $Q(k,z)$.
- Above this was $P_n(k, z; \eta_n)$ but doesn't matter how $Q(\cdot)$ is obtained

Preliminary:

- Construct very fine grid $\{\tilde{k}_l, \tilde{z}_l\}_{l=1}^{\widetilde{M}}$

Step I:

- Calculate consumption using policy rule: $\widetilde{c}_l = Q(\tilde{k}_l, \tilde{z}_l)$

Step II:

- Also calculate consumption as it is *implied* by the Euler equation.
  - Use $Q(\tilde{k}_I, \tilde{z}_I)$ to calculate $\widetilde{k}' = \tilde{z}\tilde{k}_I^{\alpha} - Q(k_i, \tilde{z}_I)$
  - Use $Q(\tilde{k}', \tilde{z}')$ to calculate $\widetilde{c}' = Q(k', z')$
  - Then, get implied consumption value $\widetilde{\widetilde{c}}_I$ from

$$\left(\widetilde{\widetilde{c}}_I\right)^{-\nu} = \sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ Q(\tilde{z}_I\tilde{k}_I^{\alpha} - Q(k_I, z_I), \exp\{\rho\ln(\tilde{z}_I) + \sqrt{2}\sigma\zeta_j\})^{-\nu}\times \\ \exp\{\rho\ln(\tilde{z}_I) + \sqrt{2}\sigma\zeta_j\}\times \\ \left(\tilde{z}_I\tilde{k}_I^{\alpha} - Q(\tilde{k}_I, \tilde{z}_I)\right)^{\alpha-1} \\ \omega_j/\sqrt{\pi} \end{array} \right]$$

Step III:

- Compare and analyze differences
- Calculate maximum and average difference between $\widetilde{\widetilde{c}}_l$ and $\widetilde{c}_l$
- *Check where maximum error occurs (error on unlikely location of state space may not be that important)*

Dynamic accuracy test

- One-period accuracy test may be misleading (ignores accumulation of small errors)
- Idea of accuracy test can be extended to multiperiod setting

Step I

- Construct time series for consumption and capital using approximation: $Q(k_t, z_t)$ and $k_{t+1} = z_t k_t^{\alpha} - Q(k_t, z_t)$ for $t = 1, \cdots, T$.

Dynamic accuracy test

Step II

- Also construct time series for consumption and capital with consumption calculated from

$$
(c_t^*)^{-\nu} = \sum_{j=1}^{J} \left[\begin{array}{c} \alpha\beta\times \\ Q(z_t\,(k_t^*)^{\alpha} - Q(k_t^*, z_t), \exp\{\rho\ln(z_t) + \sqrt{2}\sigma\zeta_j\})^{-\nu}\times \\ \exp\{\rho\ln(z_t) + \sqrt{2}\sigma\zeta_j\}\times \\ \left(z_t\,(k_t^*)^{\alpha} - Q(k_t^*, z_t)\right)^{\alpha-1} \\ \omega_j/\sqrt{\pi} \end{array} \right]
$$

$$
k_{t+1}^* = z_t\,(k_t^*)^{\alpha} - c_t^*
$$

- $Q(\cdot)$ is only used to calculate the conditional expectation! c_t^* is the implied consumption value and k_{t+1}^* the value calculated using this implied consumption value.
- *Stare at the graph that plots both*

Dynamic accuracy test

Step III

- *Stare at the graph that plots both*