PROJECTION METHODS

Johannes Brumm

Karlsruhe Institute of Technology (KIT)

ZICE 2017

Agenda

- Setting the stage:
  - Ramsey model: Choose simple and well-known example to focus on methods rather than model
  - Euler equations, local solutions, Euler errors
- Projection methods (based mainly on Judd (1992JET)):
  - Projection methods in 3 examples: piecewise linear collocation, Chebyshev collocation, Chebyshev Galerkin
  - Projection methods in general: mathematical formulation, cooking recipe, higher dimensions

The Deterministic Ramsey Model

Choose $\{c_t, k_{t+1}\}_{t=0}^{\infty}$ to maximize $U(\{c_t\}_{t=0}^{\infty})$ subject to

$$\forall t \in \mathbb{N}_0 : 0 \leq k_{t+1} \leq \underbrace{f(k_t) + (1-\delta)k_t}_{\equiv \bar{f}(k_t)} - c_t, 0 \leq c_t, k_0 \text{ given}$$

where:

- c_t is consumption at time t
- $U(\{c_t\}_{t=0}^{\infty})$ is utility of the consumption stream $\{c_t\}_{t=0}^{\infty}$
- k_t is the capital stock at time t, and k_0 the initial capital stock
- $f(\cdot)$ is the production function
- $\bar{f}(\cdot)$ is production including non-depreciated capital
- δ is depreciation

Standard Assumptions on Preferences and Production

Production

- Neoclassical Production:
  $f(0) = 0$, $f \in C^2(\mathbb{R})$,
  $f'(k) > 0$, $f''(k) < 0$,
  $\lim_{k\to 0} f'(k) = \infty$,
  $\lim_{k\to\infty} f'(k) = 0$
- Special Case:

$$f(k) = k^\alpha$$

  Cobb-Douglas with capital share α and fixed labor supply (normalized or intensive form)

Preferences

- Time-separable utility:

$$U(\{t\}_{t=0}^{\infty}) = \sum_{t=0}^{\infty} \beta^t u(c_t)$$

  with discount factor $0 < \beta < 1$,
  $u'(c) > 0$, $u''(c) < 0$, and
  $\lim_{c\to 0} u'(c) = \infty$.
- Special Case:

$$u(c_t) = \begin{cases} \ln(c_t), \ \gamma = 1 \\ \frac{c_t^{1-\gamma}}{1-\gamma}, \ \gamma \in \mathbb{R}_+ \setminus \{1\} \end{cases}$$

  CRRA utility

The Euler Equation

Due to the above assumptions:

- $c_t \geq 0, \; k_{t+1} \geq 0$ are never binding
- the budget constraint is always binding: $c_t = \bar{f}(k_t) - k_{t+1}$

Therefore, the **Lagrangian** of the maximization problem simplifies to:

$$\mathcal{L} = \sum_{t=0}^{\infty} \beta^t [u(c_t) + \lambda_t(\bar{f}(k_t) - c_t - k_{t+1})]$$

$$\frac{\partial \mathcal{L}}{\partial c_t} = 0 \Leftrightarrow u'(c_t) = \lambda_t; \; \frac{\partial \mathcal{L}}{\partial k_{t+1}} = 0 \Leftrightarrow \lambda_t = \beta \lambda_{t+1} \bar{f}'(k_{t+1})$$

Combining, we get the **Euler equation**(s):

$$u'(\bar{f}(k_t) - k_{t+1}) = \beta \bar{f}'(k_{t+1}) u'(\bar{f}(k_{t+1}) - k_{t+2}) \quad \forall t \in \mathbb{N}_0$$

Recursive Equilibrium

Hard to solve for an infinite sequence directly!

$\Rightarrow$ Reduce problem to two periods: 'today' and 'tormorrow'

$\Rightarrow$ Suppose optimal choice does not depend on t directly, just on k_t

$\Rightarrow$ Look for recursive equilibrium with capital k as endogenous state

$\Rightarrow$ A recursive equilibrium consumption function $C(k)$ must satisfy:

$$u'(C(k)) = \beta \cdot \bar{f}'(\bar{f}(k) - C(k)) \cdot u'(C(\bar{f}(k) - C(k)))$$

The Steady State

In a steady state, k^*, capital does not change from 'today' to 'tormorrow':

$$\bar{f}(k^*) - C(k^*) = k^*$$

This requirement and the Euler equation determine the steady state:

$$\begin{aligned} u'(C(k^*)) &= \beta \cdot \bar{f}'(k^*) \cdot u'(C(k^*)) \\ 1 &= \beta \cdot \bar{f}'(k^*) \\ k^* &= (\bar{f}')^{-1}(\tfrac{1}{\beta}) \end{aligned}$$

and therefore:

$$c^* = C(k^*) = \bar{f}(k^*) - k^*$$

Linear Approximation around the steady state

Euler equation at the Steady State:

$$u'(C(k^*)) = \beta u'(C(\bar{f}(k^*) - C(k^*)) \cdot \bar{f}'(\bar{f}(k^*) - C(k^*))$$

Differentiate with respect to $k = k^*$ and drop all arguments:

$$u'' \cdot C' = \beta u'' \cdot C'(\bar{f}' - C')\bar{f}' + \beta u' \bar{f}''(\bar{f}' - C')$$

Use that $\bar{f}' = \frac{1}{\beta}$ at the steady state and devide by u'':

$$0 = \underbrace{1}_{a} \cdot (C')^2 + \underbrace{(1 - \bar{f} + \beta \frac{u'}{u''} \bar{f}'')}_{b} C' - \underbrace{\frac{u'}{u''} \bar{f}''}_{c}$$

$$\Rightarrow C'(k^*) = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

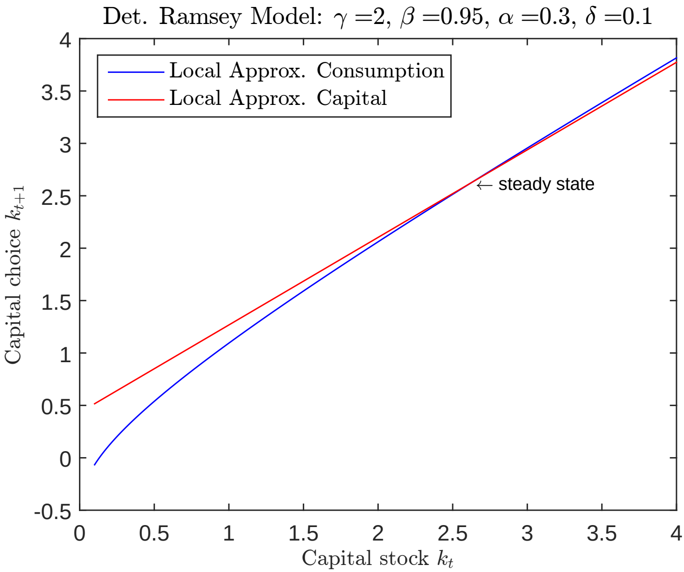
From this we get the approximations:

$$\hat{C}(k) = c^* + (k - k^*) \cdot C'(k^*)$$

$$\text{OR: } \hat{K}^+(k) = k^* + (k - k^*) \cdot (1 - C'(k^*))$$

(See Judd 1998)

Linear Approximations around the Steady State



Assessing Accuracy: Euler Errors I

We want a policy function $C(\cdot)$ that satisfies the Euler equation

$$u'(C(k)) = \beta \cdot \bar{f}'(\bar{f}(k) - C(k)) \cdot u'(C(\bar{f}(k) - C(k)))$$

at all $k \in [k_{min}, k_{max}]$, not only at k^*. We proceed as follows:

- Create many points $\{\tilde{k}_i\}_{i=1}^{I} : \tilde{k}_i \in [k_{min}, k_{max}]$
- Compute consumption implied by approximate policy: $\hat{c}_i = \hat{C}(\tilde{k}_i)$.
- Compute consumption implied by Euler equation and approximate policy 'tomorrow': $c_i^* = u_c^{-1}\left[\beta \bar{f}'(\bar{f}(\tilde{k}_i) - \hat{c}_i) \cdot u_c\left(\hat{C}(\bar{f}(\tilde{k}_i) - \hat{c}_i)\right)\right]$
- The (relative) error that the agent makes 'today' given his choice 'tomorrow' is the Euler error:

$$E_i = |\frac{\hat{c}_i}{c_i^*} - 1|$$

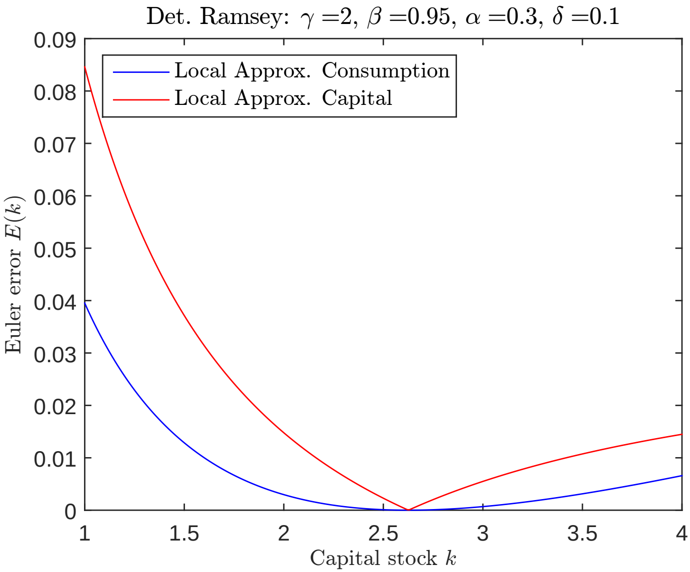
Assessing Accuracy: Euler Errors II

- Choose points $\{\tilde{k}_i\}_{i=1}^{I}$ either
  - randomly (uniformly distributed) in $[k_{min}, k_{max}]$, or
  - as a very fine (equidistant) grid on $[k_{min}, k_{max}]$
  - or along a simulated path
- 'Bounded rationality' interpretation: The Euler error

$$E_i = \left|\frac{\hat{c}_i}{c_i^*} - 1\right|$$

is the fraction by which the approximate consumption choice today differs from the optimal one (given the approximate consumption choice tomorrow). For instance, $E_i = 0.05$ means that consumption is 5% too high or too low relative to the optimum

Euler Errors for Local Approximations



Our 1^{st} Global Solution: Piecewise Linear Collocation

- The accuracy of the local solution is high close to the steady state, yet low further away
- We would like to force the solution to be accurate also further away from the steady state
- We demand that the solution satisfies the Euler equation exactly on a grid of points (instead of only at the steady state)
- As a start, we interpolate linearly between these points

Algorithm for Collocation with Piecewise Linear Basis

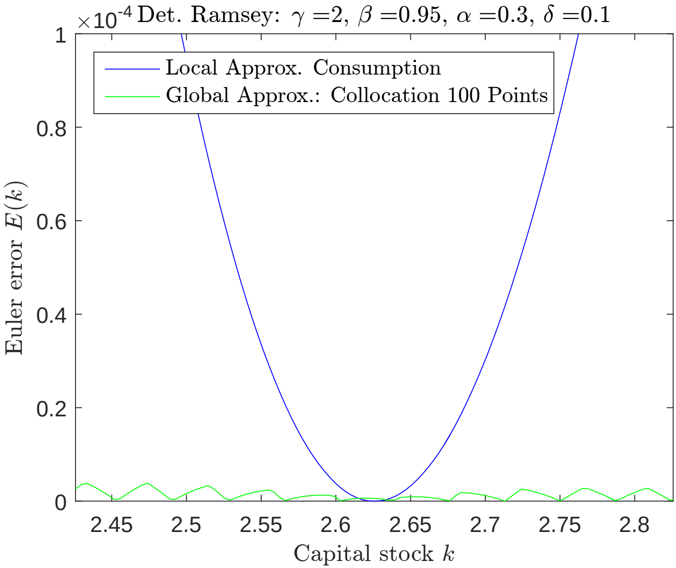
0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set capital grid $K = [K_1 \ K_2 \ \dots \ K_n] \in \mathbb{R}^n_+, \ K_j < K_{j+1} \ \forall \ j$
      and set guess for policy at gridpoints $K^+ = [K_1^+ \ K_2^+ \ \dots \ K_n^+] \in \mathbb{R}^n_+$
   b) Set error tolerance $\bar{\epsilon} > 0$
1. Main Step *(Solve for parameters of policy function)*
   Solve for K^+ the system of non-linear equations, $\forall 1 \leq j \leq n$:

$$R(K_j) \equiv u'(\bar{f}(K_j) - K_j^+) - \beta \cdot \bar{f}'(K_j^+) \cdot u' \left(\bar{f}(K_j^+) - PL(K_j^+; K, K^+) \right) = 0,$$

where $PL(x; K, K^+) \equiv \frac{(K_{j+1}^+ - x) K_j^+ + (x - K_j^+) K_{j+1}^+}{K_{j+1}^+ - K_j^+}$ for $K_j \leq x \leq K_{j+1}$

2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \max_j R_j$
   b) If $\epsilon < \bar{\epsilon}$, then stop and report result (K, K^+); otherwise go to step 0 and make different choices.

Comparing Global and Local Solution: Euler Errors



Can We Do Better?—Chebyshev Polynomials

- The policy function we are interpolating seems very smooth, thus polynomials should do much better than linear splines
- Choose basis of orthogonal polynomials $\rightarrow$ Chebyshev polynomials

Recall:

- Chebyshev polynomials: $T_n(x) \equiv \cos\left(n \arccos(x)\right)$, for $n \geq 0$
- Chebyshev zeros of T_n: $Z_j^n \equiv -cos\left(\frac{2j-1}{2n}\pi\right)$, for $j \geq 1$:
- The degree $n-1$ Chebyshev interpolant of a function g (on $[-1,1]$):

$$CP(x;a) \equiv \sum_{l=0}^{n-1} a_l T_l(x), \text{where } a_l \equiv \frac{\sum_{j=1}^{n} g(Z_j^n) T_l(Z_j^n)}{\sum_{j=1}^{n} T_l(Z_j^n)^2}$$

Algorithm for Collocation with Chebyshev Polynomials

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set Chebyshev capital grid $K = [K_1 \ K_2 \ \dots \ K_n] \in \mathbb{R}^n_+$ by choosing $n, \bar{K}, \underline{K}$ and setting $K_j = \left(Z_j^n + 1\right)\left(\bar{K} - \underline{K}\right)/2 + \underline{K}$ and set guess for parameters of policy function $a \in \mathbb{R}^n$
   b) Set error tolerance $\bar{\epsilon} > 0$
1. Main Step *(Solve for parameters of policy function)*
   Solve for a the system of non-linear equations, $\forall 1 \leq j \leq n$:

$$R(K_j; a) \equiv u'(\bar{f}(K_j) - K_j^+) - \beta \cdot \bar{f}'(K_j^+) \cdot u'\left(\bar{f}(K_j^+) - K_j^{++}\right) = 0,$$

   where

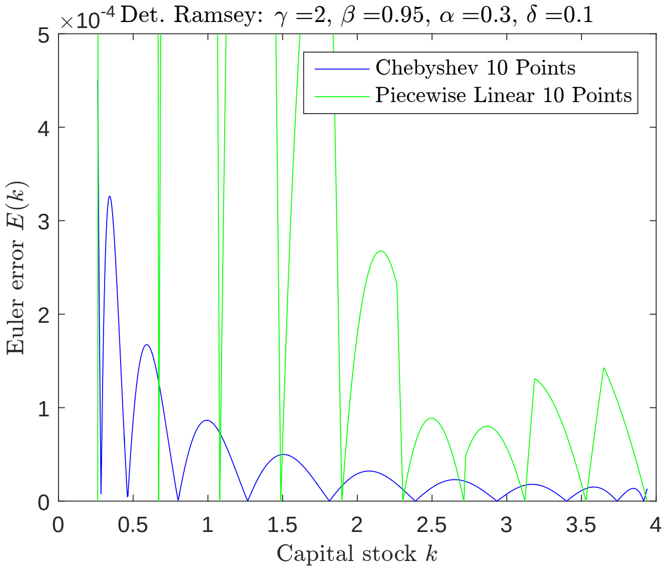
$$K_j^+ = CP\left(\rho(K_j); a\right), K_j^{++} = CP\left(\rho(K_j^+); a\right), \rho(x) = \frac{2(x - \underline{K})}{\bar{K} - \underline{K}} - 1$$

2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \max_j R_j$
   b) If $\epsilon < \bar{\epsilon}$, then stop and report a; otherwise go to step 0.

Chebyshev Coefficients

coefficient	$n = 10, \gamma = 2$	$n = 10, \gamma = 0.1$	$n = 5, \gamma = 0.1$
a_0	+2.1370	+2.2142	+2.2133
a_1	+1.6658	+1.1781	+1.1762
a_2	-0.0240	-0.1234	-0.1221
a_3	+0.0093	+0.0357	+0.0360
a_4	-0.0039	-0.0130	-0.0112
a_5	+0.0018	+0.0053	
a_6	-0.0008	-0.0023	
a_7	+0.0004	+0.0011	
a_8	-0.0002	-0.0005	
a_9	+0.0001	+0.0002	

Comparing Piecewise Linear and Chebyshev Interpolation



Collocation—A Special Case of Projection Methods

- As above, let $R(x; a)$ denote the Euler Equation residual at x for the Chebyshev polynomial with coefficients a
- Then the Chebyshev collocation method solves for a such that

$$\forall 1 \leq j \leq n : R(K_j; a) = 0$$

- Using the Dirac delta "function" this can be written as:

$$\forall 1 \leq j \leq n : P_j^C(a) \equiv \int_{\underline{K}}^{\bar{K}} R(x; a) \cdot \delta(x - K_j) dx = 0$$

- In general, we weight the residual with functions $\{w_j\}_{j=1\ldots n}$ and aim for all these "projections" to become zero:

$$\forall 1 \leq j \leq n : P_j(a) \equiv \int_{\underline{K}}^{\bar{K}} R(x; a) w_j(x) dx = 0$$

Chebyshev Galerkin Method

- Galerkin idea: Use the first n elements of the basis as projections
- Thus, with Chebyshev basis, choose in the above formula:

$$w_j(x) = T_j(\rho(x)) \frac{1}{\sqrt{1-\rho(x)^2}}, \text{where } \rho(x) = \frac{2(x-\underline{K})}{\bar{K}-\underline{K}} - 1$$

- Thus we have to solve for a such that

$$\forall 1 \leq j \leq n : P_j(a) \equiv \int_{\underline{K}}^{\bar{K}} R(x;a) T_j(\rho(x)) \frac{1}{\sqrt{1-\rho(x)^2}} dx = 0$$

- However, these integrals have to be approximated. Using Gauss-Chebyshev quadrature and $m > n$ (below we have $m = 2n+1$) we get:

$$\forall 1 \leq j \leq n : \sum_{l=1}^{m} R(K_l;a) T_j(\rho(K_l)) = 0$$

where now $K_l = (Z_l^m + 1)\left(\bar{K} - \underline{K}\right)/2 + \underline{K}$

Comparing Projection Methods

	1^{st} Example	2^{nd} Example	3^{rd} Example
Method	Collocation		Galerkin
Basis Function	Piecewise Linear	Chebyshev	Chebyshev
Weighting	Dirac measure at gridpoints		Chebyshev
Max/Avg EE 4 Points	2.7(-2), 6.6(-3)	2.2(-2), 2.0(-3)	8.0(-3), 3.1(-3)
Max/Avg EE 16 Points	1.2(-2), 8.0(-4)	1.1(-5), 9.8(-7)	4.3(-6), 1.5(-6)
Max/Avg EE 64 Points	7.3(-4), 3.2(-5)	$<<$ 1(-10)	$<<$ 1(-10)

Projection Methods: The General Problem Statement

- We look for the solution g of an operator equation:
$$\mathcal{R}(g) = 0,$$
where the operator $\mathcal{R} : B \to B$ is a self-mapping on a function space (for simplicity we do not consider $\mathcal{R} : B_1 \to B_2$ with $B_1 \neq B_2$)
- In our example, $\mathcal{R}$ operates on $C^0([\underline{K}, \bar{K}], \mathbb{R})$ mapping a policy function $p : [\underline{K}, \bar{K}] \to \mathbb{R}$ into the residual function $R : [\underline{K}, \bar{K}] \to \mathbb{R}$ with
$$R(\cdot; p) = u'(\bar{f}(\cdot) - p(\cdot)) - \beta \cdot \bar{f}'(p(\cdot)) \cdot u'\left(\bar{f}(p(\cdot)) - p(p(\cdot))\right)$$
Thus, the solution to the operator equation is the policy function p which makes the residual function $R(\cdot; p)$ equal to the zero function
- Another example is $\mathcal{R}(v) = T(v) - v$, with the Bellman operator T operating on a space of value functions

A Cooking Recipe for Projection Methods

1. Choose basis $\{b_l\}_{l=0}^{\infty}$ and inner product $\langle \cdot, \cdot \rangle$ for the function space B
2. Choose a degree of approximation n for interpolation, and n projection functions $\{p_j\}_{j=1}^{n}$
3. For a guess for the coefficients a define the approximation and the residual function:

$$\hat{g}(x) \equiv \sum_{l=0}^{n-1} a_l b_l(x), \quad R(x; a) \equiv (\mathcal{R}(\hat{g}))(x)$$

4. Find coefficients a that solve

$$\forall 1 \leq j \leq n : \langle R(\cdot; a), p_j(\cdot) \rangle = 0$$

Going to Higher Dimensions

- In principle, the collocation method and the Galerkin method both have straightforward extensions to higher dimensions through tensor product constructions
- However, both suffer from the curse of dimensionality
- To overcome this curse,
  - in the Galerkin method: First, Gauss-Chebyshev integration can be replaced by non-product monomial rules; second, tensor products of Chebyshev polynomials can be replaced by complete polynomials (see Pichler 2011JEDC)
  - in the collocation method: tensor product grids can be replaced by sparse grids (see Krueger and Kubler 2004JEDC) or adaptive sparse grids (see Brumm and Scheidegger 2016WP)

Discussion

- Projection methods are a powerful tool to compute global solutions of dynamic economic models
- Using projection methods you can take advantage of modern non-linear equation solvers
- However, convergence is in many applications not guaranteed. Therefore:
  - Find good starting guesses
  - Try different solvers and their options
  - Change degree and type of basis functions, try different projection functions
  - Or: Resort to dynamic programming or time iteration ...