ML-3
7/22/11

Projection Methods

Jesús Fernández-Villaverde

University of Pennsylvania

July 10, 2011

Introduction

- We come back to our functional equation:

$$\mathcal{H}(d) = \mathbf{0}$$

- Projection methods solve the problem by specifying:

$$d^n(x, \theta) = \sum_{i=0}^{n} \theta_i \Psi_i(x)$$

We pick a basis $\{\Psi_i(x)\}_{i=0}^{\infty}$ and "project" $\mathcal{H}(\cdot)$ against that basis to find the θ_i's.

- How?

Points to Emphasize

1. We may want to approximate different objects d: for instance a decision rule, a value function, or an expectation.

2. In general we will have with the same number of parameters than basis functions.

3. We will work with linear combinations of basis functions. Why? The theory of nonlinear approximations is not yet as developed as the linear case.

Basic Algorithm

1. Define n known linearly independent functions $\psi_i : \Omega \longrightarrow \Re^m$ where $n < \infty$. We call the $\psi_1(\cdot), \psi_2(\cdot), ..., \psi_n(\cdot)$ the *basis functions*.
2. Define a vector of parameters $\theta = [\theta_1, \theta_2, ..., \theta_n]$.
3. Define a combination of the basis functions and the θ's:

$$d^n(\cdot|\theta) = \sum_{i=1}^{n} \theta_i \psi_n(\cdot)$$

4. Plug $d^n(\cdot|\theta)$ into $H(\cdot)$ to find the *residual equation*:

$$R(\cdot|\theta) = \mathcal{H}(d^n(\cdot|\theta))$$

5. Find the value of $\widehat{\theta}$ that make the residual equation as close to $\mathbf{0}$ as possible given some objective function $\rho : J^1 \times J^1 \longrightarrow J^2$:

$$\widehat{\theta} = \arg\min_{\theta \in \Re^n} \rho(R(\cdot|\theta), \mathbf{0})$$

Relation with Econometrics

- Looks a lot like OLS. Explore this similarity later in more detail.
- Also with semi-nonparametric methods as Sieves.
- Compare with:
  1. Policy iteration.
  2. Parameterized Expectations.

Two Issues

We need to decide:

1. Which basis we use?
    1. Pick a global basis⇒spectral methods.
    2. Pick a local basis⇒finite elements methods.
2. How do we "project"?

Different choices in 1 and 2 will result in slightly different projection methods.

Spectral Methods

- Main reference: Judd (1992).
- Spectral techniques use basis functions that are nonzero and smooth almost everywhere in Ω.
- Advantages: simplicity.
- Disadvantages: difficult to capture local behavior. Gibbs phenomenon.

Spectral Basis I

Monomials: $c, x, x^2, x^3, ...$

- Simple and intuitive.
- Even if this basis is not composed by orthogonal functions, if J_1 is the space of bounded measurable functions on a compact set, the Stone-Weierstrass theorem assures completeness in the L^1 norm.
- Problems:
  1. (Nearly) multicollinearity. Compare the graph of x^{10} with x^{11}. The solution of a projection involves matrices inversion. When the basis functions are similar, the condition number of these matrices (the ratio of the largest and smallest absolute eigenvalues) are too high. Just the six first monomials can generate conditions numbers of 10^{10}. The matrix of the LS problem of fitting a polynomial of degree 6 to a function (the *Hilbert Matrix)*, is a popular test of numerical accuracy since it maximizes rounding errors!
  2. Monomials vary considerably in size, leading to scaling problems and accumulation of numerical errors.
- We want an orthogonal basis.

Spectral Basis II

Trigonometric series

$$1/\left(2\pi\right)^{0.5}, \cos x/\left(2\pi\right)^{0.5}, \sin x/\left(2\pi\right)^{0.5}, ...,$$
$$\cos kx/\left(2\pi\right)^{0.5}, \sin kx/\left(2\pi\right)^{0.5}, ...$$

- Periodic functions.
- However economic problems are generally not periodic.
- Periodic approximations to nonperiodic functions suffer from the Gibbs phenomenon, requiring many terms to achieve good numerical performance (the rate of convergence to the true solution as $n \longrightarrow \infty$ is only $O\left(n\right)$).

Spectral Basis III

- Flexible class: orthogonal polynomials of Jacobi (or hypergeometric) type. Why orthogonal?

- The Jacobi polynomial of degree n, $P_n^{\alpha,\beta}(x)$ for $\alpha, \beta > -1$, is defined by the orthogonality condition:

$$\int_{-1}^{1} (1-x)^{\alpha} (1+x)^{\beta} P_n^{\alpha,\beta}(x) P_m^{\alpha,\beta}(x) dx = 0 \text{ for } m \neq n$$

- The two most important cases of Jacobi polynomials:

  1. Legendre: $\alpha = \beta = -\frac{1}{2}$.

  2. Chebyshev: $\alpha = \beta = 0$.

Alternative Expressions

- The orthogonality condition implies, with the customary normalizations:

$$P_n^{\alpha,\beta}(1) = \binom{n+\alpha}{n}$$

that the general n term is given by:

$$2^{-n} \sum_{k=0}^{n} \binom{n+\alpha}{k} \binom{n+\beta}{n-k} (x-1)^{n-k} (x+1)^k$$

- Recursively:

$$\begin{gathered} 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta) P_{n+1} = \\ \left(\begin{array}{c} (2n+\alpha+\beta+1)(\alpha^2-\beta^2) \\ +(2n+\alpha+\beta)(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)x \end{array} \right) P_n \\ -2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2) P_{n-1} \end{gathered}$$

Chebyshev Polynomials

- One of the most common tools of Applied Mathematics.
- References:
  - *Chebyshev and Fourier Spectral Methods*, John P. Boyd (2001).
  - *A Practical Guide to Pseudospectral Methods*, Bengt Fornberg (1998).
- Advantages of Chebyshev Polynomials:
  1. Numerous simple close-form expressions are available.
  2. The change between the coefficients of a Chebyshev expansion of a function and the values of the function at the Chebyshev nodes are quickly performed by the cosine transform.
  3. They are more robust than their alternatives for interpolation.
  4. They are bounded between $[-1, 1]$ while Legendre polynomials are not, offering a better performance close to the boundaries of the problems.
  5. They are smooth functions.
  6. Several theorems bound the errors for Chebyshev polynomials interpolations.

Definition of Chebyshev Polynomials I

- Recursive definition:

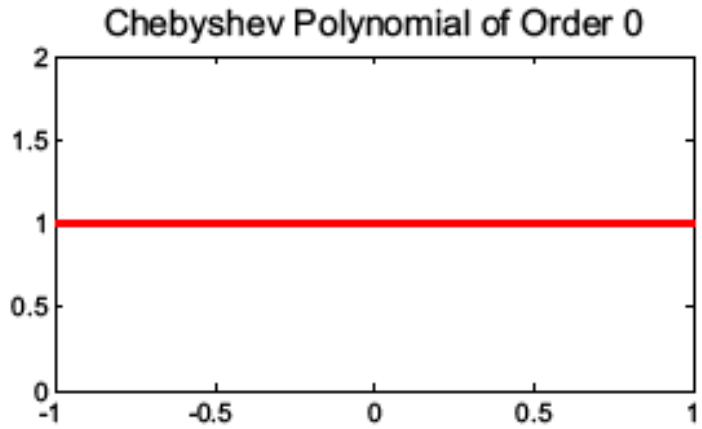
$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x) \text{ for a general } n \end{aligned}$$

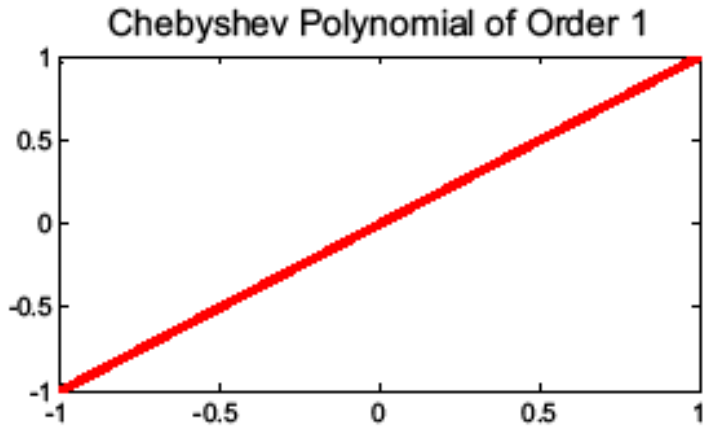
- The first few polynomials are then 1, x, $2x^2 - 1$, $4x^3 - 3x$, $8x^4 - 8x^2 + 1$, etc...

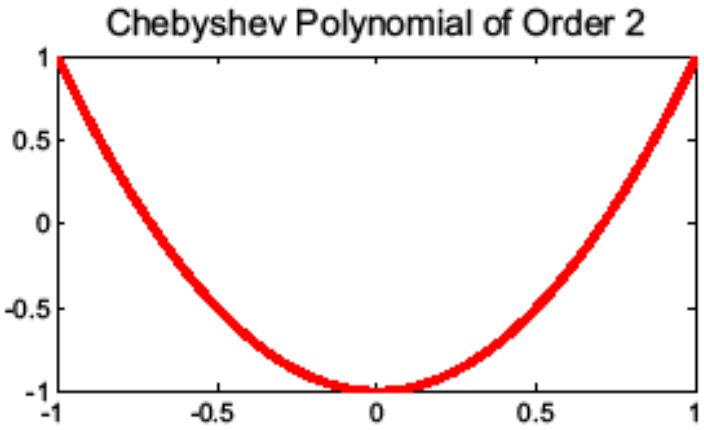
- The n zeros of the polynomial $T_n(x_k) = 0$ are given by:

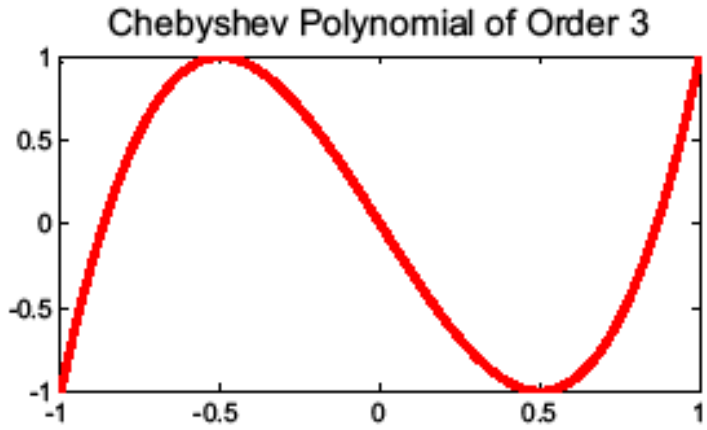
$$x_k = \cos \frac{2k-1}{2n}\pi, k = 1, ..., n$$

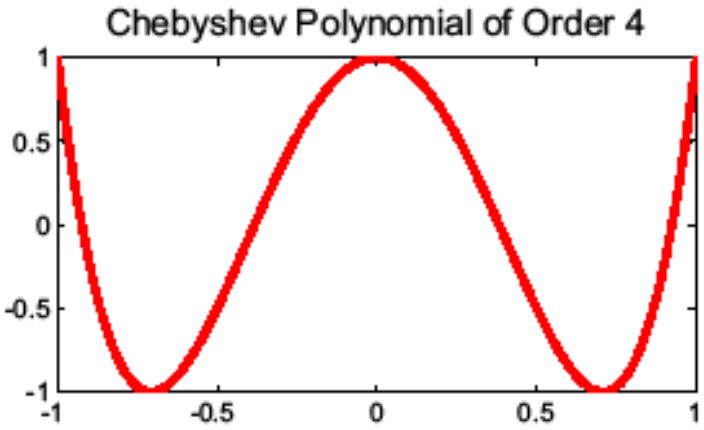
- Note that zeros are clustered quadratically towards ± 1.

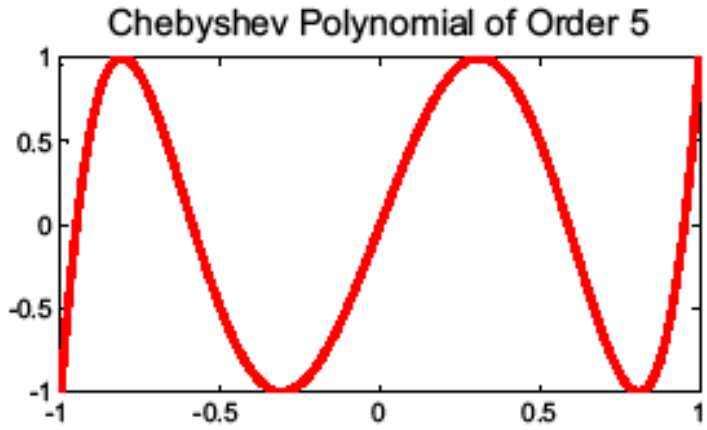

Chebyshev Polynomial of Order 0
2
1.5
1
0.5
0
-1
-0.5
0
0.5
1


Chebyshev Polynomial of Order 1
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


Chebyshev Polynomial of Order 2
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


Chebyshev Polynomial of Order 3
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


Chebyshev Polynomial of Order 4
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


Chebyshev Polynomial of Order 5
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

Definition of Chebyshev Polynomials II

- Explicit definition:

$$\begin{aligned}
T_n(x) &= \cos(n \arccos x) \\
&= \frac{1}{2}\left(z^n + \frac{1}{z^n}\right) \text{ where } \frac{1}{2}\left(z + \frac{1}{z}\right) = x \\
&= \frac{1}{2}\left(\left(x + \left(x^2 - 1\right)^{0.5}\right)^n + \left(x - \left(x^2 - 1\right)^{0.5}\right)^n\right) \\
&= \frac{1}{2}\sum_{k=0}^{[n/2]} (-1)^k \frac{(n-k-1)!}{k!\,(n-2k)!}(2x)^{n-2k} \\
&= \frac{(-1)^n \pi^{0.5}}{2^n \Gamma\left(n + \frac{1}{2}\right)}\left(1 - x^2\right)^{0.5} \frac{d^n}{dx^n}\left(\left(1 - x^2\right)^{n - \frac{1}{2}}\right)
\end{aligned}$$

Remarks

- The domain of the Chebyshev polynomials is $[-1,1]$. Since our state space is, in general, different, we use a linear translation from $[a,b]$ into $[-1,1]$:

$$2\frac{x-a}{b-a}-1$$

- Chebyshev polynomials are orthogonal with respect to the weight function:

$$\frac{1}{\left(1-x^2\right)^{0.5}}$$

Chebyshev Interpolation Theorem

if an approximating function is exact at the roots of the n_1^{th} order Chebyshev polynomial then, as $n_1 \longrightarrow \infty$, the approximation error becomes arbitrarily small.

Multidimensional Problems

- Chebyshev polynomials are defined on $[-1, 1]$.
- However, most problems in economics are multidimensional.
- How do we generalize the basis?
- Curse of dimensionality.

Tensors

- Assume we want to approximate $F : [-1, 1]^d \to \mathbb{R}$.
- Let T_j denote the Chebyshev polynomial of degree $j = 0, 1, .., \kappa$.
- We can approximate F with tensor product of Chebyshev polynomials of degree κ:

$$\hat{F}(x) = \sum_{n_1=0}^{\kappa} \cdots \sum_{n_d=0}^{\kappa} \xi_{n_1,\ldots,n_d} T_{n_1}(x_1) \cdots T_{n_d}(x_d)$$

- Beyond simplicity, an advantage of the tensor basis is that if the one-dimensional basis is orthogonal in a norm, the tensor basis is orthogonal in the product norm.
- Disadvantage: number of elements increases exponentially. We end up having terms $x_1^\kappa x_2^\kappa \cdots x_d^\kappa$, total number of $(\kappa + 1)^d$.

Complete Polynomials

- Solution: eliminate some elements of the tensor in such a way that there is not much numerical degradation.

- Judd and Gaspar (1997): Use complete polynomials instead

$$\mathcal{P}_{\kappa}^{d} \equiv \left\{ x_1^{i_1} \cdots x_d^{i_d} \text{ with } \sum_{l=1}^{d} i_l \leq \kappa,\ 0 \leq i_1, ..., i_d \right\}$$

- Advantage: much smaller number of terms, no terms of order $d\kappa$ to evaluate.

- Disadvantage: still too many elements.

Smolyak's Algorithm I

- Define $m_1 = 1$ and $m_i = 2^{i-1} + 1$, $i = 2, ...$
- Define $G^i = \{x_1^i, ..., x_{m_i}^i\} \subset [-1, 1]$ as the set of the extrema of the Chebyshev polynomials

$$x_j^i = -cos\left(\frac{\pi(j-1)}{m_i - 1}\right) \quad j = 1, ..., m_i$$

with $G^1 = \{0\}$. It is crucial that $G^i \subset G^{i+1}$, $\forall i = 1, 2, \ldots$

- Example:

$$\begin{aligned} i &= 1, m_i = 1, \mathcal{G}^i = \{0\} \\ i &= 2, m_i = 3, \mathcal{G}^i = \{-1, 0, 1\} \\ i &= 3, m_i = 5, \mathcal{G}^i = \left\{-1, -\cos\left(\frac{\pi}{4}\right), 0, -\cos\left(\frac{3\pi}{4}\right), 1\right\} \end{aligned}$$

Smolyak's Algorithm II

- For $q > d$, define a sparse grid

$$\mathcal{H}(q,d) = \bigcup_{q-d+1 \leq |i| \leq q} (\mathcal{G}^{i_1} \times ... \times \mathcal{G}^{i_d}),$$

where $|i| = i_1 + \ldots + i_d$. The number q defines the size of the grid and thus the precision of the approximation.

- For example, let $q = d + 2 = 5$:

$$\mathcal{H}(5,3) = \bigcup_{3 \leq |i| \leq 5} (\mathcal{G}^{i_1} \times ... \times \mathcal{G}^{i_d}).$$

$$\begin{array}{lll}
\mathcal{G}^3 \times \mathcal{G}^1 \times \mathcal{G}^1, & \mathcal{G}^1 \times \mathcal{G}^3 \times \mathcal{G}^1, & \mathcal{G}^1 \times \mathcal{G}^1 \times \mathcal{G}^3 \\
\mathcal{G}^2 \times \mathcal{G}^2 \times \mathcal{G}^1, & \mathcal{G}^2 \times \mathcal{G}^1 \times \mathcal{G}^2, & \mathcal{G}^1 \times \mathcal{G}^2 \times \mathcal{G}^2 \\
\mathcal{G}^2 \times \mathcal{G}^1 \times \mathcal{G}^1, & \mathcal{G}^1 \times \mathcal{G}^2 \times \mathcal{G}^1, & \mathcal{G}^1 \times \mathcal{G}^1 \times \mathcal{G}^2 \\
\mathcal{G}^1 \times \mathcal{G}^1 \times \mathcal{G}^1 & &
\end{array}$$

Smolyak's Algorithm III

- Number of points for $q = d + 2$

$$1 + 4d + 4\frac{d(d-1)}{2}$$

- Largest number of points along one dimension

$$\begin{aligned} i &= q - d + 1 \\ m_i &= 2^{q-d} + 1 \end{aligned}$$

- Rectangular grid

$$\left[2^{q-d} + 1\right]^d$$

- Key: with rectangular grid, the number of grid points increases exponentially in the number of dimensions. With the Smolyak algorithm number of points increases polynomially in the number of dimensions.

Smolyak's Algorithm IV

Size of the Grid for $q = d + 2$			
d	$2^{q-d} + 1$	$\#\mathcal{H}(q, d)$	$\left[2^{q-d} + 1\right]^d$
2	5	13	25
3	5	25	125
4	5	41	625
5	5	61	3,125
12	5	313	244,140,625

Smolyak's Algorithm V

- For one dimension denote the interpolating Chebyshev polynomials as

$$\mathcal{U}^i(x^i) = \sum_{j=1}^{m_i} \xi_j^i T_j(x^i)$$

  and the d-dimensional tensor product by $U^{i_1} \otimes ... \otimes U^{i_d}(x)$.
- For $q > d$, approximating function (Smolyak's algorithm) given by

$$\mathcal{A}(q,d)(x) = \sum_{q-d+1 \leq |i| \leq q} (-1)^{q-|i|} \begin{pmatrix} d-1 \\ q-|i| \end{pmatrix} (\mathcal{U}^{i_1} \otimes \ldots \otimes \mathcal{U}^{i_d})(x)$$

- Method is (almost) optimal within the set of polynomial approximations (Barthelmann, Novak, and Ritter, 1999).
- Method is universal, that is, almost optimal for many different function spaces.

Boyd's Moral Principal

1. When in doubt, use Chebyshev polynomials unless the solution is spatially periodic, in which case an ordinary Fourier series is better.
2. Unless you are sure another set of basis functions is better, use Chebyshev polynomials.
3. Unless you are really, really sure another set of basis functions is better, use Chebyshev polynomials.

Finite Elements

- Standard Reference: McGrattan (1999).
- Bound the domain Ω in small of the state variables.
- Partition Ω in small in nonintersecting elements.
- These small sections are called elements.
- The boundaries of the elements are called nodes.

Partition into Elements

- Elements may be of unequal size.
- We can have small elements in the areas of Ω where the economy will spend most of the time while just a few, big size elements will cover wide areas of the state space infrequently visited.
- Also, through elements, we can easily handle issues like kinks or constraints.
- There is a whole area of research concentrated on the optimal generation of an element grid. See Thomson, Warsi, and Mastin (1985).

Structure

- Choose a basis for the policy functions in each element.
- Since the elements are small, a linear basis is often good enough:

$$\psi_i(k) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if } k \in [x_i, x_{i+1}] \\ 0 \text{ elsewhere} \end{cases}$$

- Plug the policy function in the Equilibrium Conditions and find the unknown coefficients.
- Paste it together to ensure continuity.
- Why is this an smart strategy?
- Advantages: we will need to invert an sparse matrix.
- When should be choose this strategy? speed of computation versus accuracy.

Three Different Refinements

1. *h-refinement*: subdivide each element into smaller elements to improve resolution uniformly over the domain.

2. *r-refinement*: subdivide each element only in those regions where there are high nonlinearities.

3. *p-refinement*: increase the order of the approximation in each element. If the order of the expansion is high enough, we will generate in that way an hybrid of finite and spectral methods knows as spectral elements.

Choosing the Objective Function

- The most common answer to the second question is given by a *weighted residual*.
- That is why often projection methods are also called weighted residual methods
- This set of techniques propose to get the residual close to $\mathbf{0}$ in the weighted integral sense.
- Given some weight functions $\phi_i : \Omega \to \Re^m$:

$$\rho\left(R\left(\cdot|\,\theta\right),\mathbf{0}\right) = \begin{cases} 0 \text{ if } \int_\Omega \phi_i\left(x\right) R\left(\cdot|\,\theta\right) dx = \mathbf{0},\ i = 1,..,n \\ 1 \text{ otherwise} \end{cases}$$

- Then the problem is to choose the θ that solve the system of equations:

$$\int_\Omega \phi_i\left(x\right) R\left(\cdot|\,\theta\right) dx = \mathbf{0},\ i = 1,..,n$$

Remarks

- With the approximation of d by some functions ψ_i and the definition of some weight functions $\phi_i(\cdot)$, we have transform a rather intractable functional equation problem into the standard nonlinear equations system!
- The solution of this system can be found using standard methods, as a Newton for relatively small problems or a conjugate gradient for bigger ones.
- Issue: we have different choices for an weight function:

Weight Function I: Least Squares

- $\phi_i(x) = \frac{\partial R(x|\theta)}{\partial \theta_i}$.
- This choice is motivated by the solution of the variational problem:

$$\min_{\theta} \int_{\Omega} R^2(\cdot|\theta)\, dx$$

with first order condition:

$$\int_{\Omega} \frac{\partial R(x|\theta)}{\partial \theta_i} R(\cdot|\theta)\, dx = \mathbf{0},\ i = 1, .., n$$

- Variational problem is mathematically equivalent to a standard regression problem in econometrics.
- OLS or NLLS are regression against a manifold spanned by the observations.

Weight Function I: Least Squares

- Least Squares always generates symmetric matrices even if the operator $\mathcal{H}$ is not self-adjoint.
- Symmetric matrices are convenient theoretically (they simplify the proofs) and computationally (there are algorithms that exploit their structure to increase speed and decrease memory requirements).
- However, least squares may lead to ill-conditioning and systems of equations complicated to solve numerically.

Weight Function II: Subdomain

- We divide the domain Ω in n subdomains Ω_i and define the n step functions:

$$\phi_i(x) = \begin{cases} 1 \text{ if } x \in \Omega_i \\ 0 \text{ otherwise} \end{cases}$$

- This choice is then equivalent to solve the system:

$$\int_{\Omega_i} R(\cdot|\theta)\, dx = \mathbf{0},\ i = 1, .., n$$

Weight Function III: Moments

- Take $\left\{0, x, x^2, ..., x^{n-1}\right\}$ and compute the first n periods of the residual function:

$$\int_{\Omega_i} x^i R\left(\cdot | \theta\right) dx = \mathbf{0},\ i = 0, .., n$$

- This approach, widely used in engineering works well for a low n (2 or 3).

- However, for higher orders, its numerical performance is very low: high orders of x are highly collinear and arise serious rounding error problems.

- Hence, moments are to be avoided as weight functions.

Weight Function III: Collocation or Pseudospectral or Method of Selected Points

- $\phi_i(x) = \delta(x - x_i)$ where δ is the dirac delta function and x_i are the collocation points.
- This method implies that the residual function is zero at the n collocation points.
- Simple to compute since the integral only needs to be evaluated in one point. Specially attractive when dealing with strong nonlinearities.
- A systematic way to pick collocation points is to use a density function:

$$\mu_\gamma(x) = \frac{\Gamma\left(\frac{3}{2} - \gamma\right)}{(1 - x^2)^\gamma \pi^{\frac{1}{2}} \Gamma(1 - \gamma)} \quad \gamma < 1$$

and find the collocation points as the x_j, $j = 0, ..., n-1$ solutions to:

$$\int_{-1}^{x_j} \mu_\gamma(x)\, dx = \frac{j}{n}$$

- For $\gamma = 0$, the density function implies equispaced points.

Weight Function IV: Orthogonal Collocation

- Variation of the collocation method:
  1. Basis functions are a set of orthogonal polynomials.
  2. Collocation points given by the roots of the $n-th$ polynomial.
- When we use Chebyshev polynomials, their roots are the collocation points implied by $\mu_{\frac{1}{2}}(x)$ and their clustering can be shown to be optimal as $n \longrightarrow \infty$.
- Surprisingly good performance of orthogonal collocation methods.

Weight Function V: Galerkin or Rayleigh-Ritz

- $\phi_i(x) = \psi_i(x)$ with a linear approximating function $\sum_{i=1}^{n} \theta_i \psi_i(x)$.
- Then:

$$\int_{\Omega} \psi_i(x) H\left(\sum_{i=1}^{n} \theta_i \psi_i(x)\right) dx = \mathbf{0},\ i = 1, .., n$$

  that is, the residual has to be orthogonal to each of the basis functions.
- Galerkin is a highly accurate and robust but difficult to code.
- If the basis functions are complete over J_1 (they are indeed a basis of the space), then the Galerkin solution will converge pointwise to the true solution as n goes to infinity:

$$\lim_{n \to \infty} \sum_{i=1}^{n} \theta_i \psi_i(\cdot) = d(\cdot)$$

- Experience suggests that a Galerkin approximation of order n is as accurate as a Pseudospectral $n+1$ or $n+2$ expansion.

A Simple Example

- Imagine that the law of motion for the price x of a good is given by:

$$d'(x) + d(x) = 0$$

- Let us apply a simple projection to solve this differential equation.

- Code: `test.m`, `test2.m`, `test3.m`

Analysis of Error

- As with projection, it is important to study the Euler equation errors.
- We can improve errors:
  1. Adding additional functions in the basis.
  2. Refine the elements.
- Multigrid schemes.

A More Serious Example

- Representative agent with utility function

$$U = E_0 \sum_{t=0}^{\infty} \beta^t \frac{\left(c_t^{\theta} \left(1 - l_t \right)^{1-\theta} \right)^{1-\tau}}{1-\tau}$$

- One good produced according to $y_t = e^{z_t} A k_t^{\alpha} l_t^{1-\alpha}$ with $\alpha \in (0,1)$.
- Productivity evolves $z_t = \rho z_{t-1} + \epsilon_t$, $|\rho| < 1$ and $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon)$.
- Law of motion for capital $k_{t+1} = i_t + (1-\delta) k_t$.
- Resource constrain $c_t + i_t = y_t$.

- Solve for $c(\cdot,\cdot)$ and $l(\cdot,\cdot)$ given initial conditions.

- Characterized by:

$$U_c(t) = \beta E_t \left[U_c(t+1) \left(1 + \alpha A e^{z_t+1} k_{t+1}^{\alpha-1} l(k_{t+1}, z_{t+1})^{\alpha} - \delta \right) \right]$$

$$\frac{1-\theta}{\theta} \frac{c(k_t, z_t)}{1 - l(k_t, z_t)} = (1-\alpha) e^{z_t} A k_t^{\alpha} l(k_t, z_t)^{-\alpha}$$

- A system of functional equations with no known analytical solution.

- Fortran code using Chebyshev and Finite Elements.

Chebyshev I

- We approximate the decision rules for labor as $l_t = \sum_{i=1}^{n} \theta_i \psi_i (k_t, z_t)$ where $\{\psi_i (k, z)\}_{i=1}^{n}$ are basis functions and $\theta = [\{\theta_i\}_{i=1}^{n}]$ unknown coefficients.
- We use that policy function to solve for consumption using the static first order condition.
- We build a residual function $R(k, z, \theta)$ using the Euler equation and the static first order condition.
- Then we choose θ by solving:

$$\int_{[k_{\min}, k_{\max}]} \int_{[z_{\min}, z_{\max}]} \phi_i (k, z) R(k, z, \theta) = 0 \quad \text{for } i = 1, ..., n$$

where $\{\phi_i (k, z)\}_{i=1}^{n}$ are some weight functions.

Chebyshev I

- We use a collocation method that sets $\phi_i(k,z) = \delta(k-k_j, z-z_v)$ where $\delta(.)$ is the dirac delta function, $j = 1, ..., n_1$, $v = 1, ..., n_2$ and $n = n_1 \times n_2$ and collocation points $\{k_j\}_{j=1}^{n_1}$ and $\{z_v\}_{v=1}^{n_2}$.
- For the technology shocks and transition probabilities we use Tauchen (1986)'s finite approximation to an AR(1) process and obtain n_2 points.
- We solve the system of n equations $R(k_i, z_i, \theta) = 0$ in n unknowns θ using a Quasi-Newton method.
- We use an iteration based on the increment of the number of basis functions and a nonlinear transform of the objective function (apply $(u')^{-1}$).

Finite Elements

Rewrite Euler equation as

$$U_c(k_t, z_t) = \frac{\beta}{(2\pi\sigma)^{0.5}} \int_{-\infty}^{\infty} [U_c(k_{t+1}, z_{t+1})(r(k_{t+1}, z_{t+1})] \exp(-\frac{\epsilon_{t+1}^2}{2\sigma^2}) d\epsilon_{t+1}$$

where

$$U_c(t) = U_c(k_t, z_t)$$
$$k_{t+1} = e^{z_{t+1}} k_t^{\alpha} l_t^{1-\alpha} + (1-\delta)k_t - c(k_t, z_t)$$
$$r(k_{t+1}, z_{t+1}) = 1 + \alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} l(k_{t+1}, z_{t+1})^{1-\alpha} - \delta$$

and

$$z_{t+1} = \rho z_t + \epsilon_{t+1}$$

Goal

- The problem is to find two policy functions $c(k,z) : R^+ \times [0,\infty] \longrightarrow R^+$ and $l(k,z) : R^+ \times [0,\infty] \longrightarrow [0,1]$ that satisfy the model equilibrium conditions.

- Since the static first order condition gives a relation between the two policy functions, we only need to solve for one of them.

- For the rest of the exposition we will assume that we actually solve for $l(k,z)$ and then we find $c\left(l(k,z)\right)$.

Bounding the State Space I

- We bound the domain of the state variables to partition it in nonintersecting elements.
- To bound the productivity level of the economy define $\lambda_t = \tanh(z_t)$.
- Since $\lambda_t \in [-1, 1]$ we can write the stochastic process as:

$$\lambda_t = \tanh(\rho \tanh^{-1}(z_{t-1}) + 2^{0.5}\sigma v_t)$$

where $v_t = \frac{\epsilon_t}{2^{0.5}\sigma}$.

Bounding the State Space II

- Now, since $\exp(\tanh^{-1}(z_{t-1})) = \frac{(1+\lambda_{t+1})^{0.5}}{(1-\lambda_{t+1})^{0.5}} = \widehat{\lambda}_{t+1}$, we have:

$$U_c(t) = \frac{\beta}{\pi^{0.5}} \int_{-1}^{1} \left[U_c(k_{t+1}, z_{t+1}) r(k_{t+1}, z_{t+1}) \right] \exp(-v_{t+1}^2) dv_{t+1}$$

where

$$k_{t+1} = \widehat{\lambda}_{t+1} k_t^{\alpha} l(k_t, z_t)^{1-\alpha} + (1-\delta) k_t - c\left(l(k_t, z_t)\right)$$
$$r(k_{t+1}, z_{t+1}) = 1 + \alpha \widehat{\lambda}_{t+1} k_{t+1}^{\alpha-1} l(k_{t+1}, z_{t+1})^{1-\alpha} - \delta$$

and $z_{t+1} = \tanh(\rho \tanh^{-1}(z_t) + 2^{0.5} \sigma v_{t+1})$.

- To bound the capital we fix an ex-ante upper bound $k_{\max}$, picked sufficiently high that it will only bind with an extremely low probability.

Partition into Elements

- Define $\Omega = [0, k_{\max}] \times [-1, 1]$ as the domain of $I_{fe}(k, z; \theta)$.
- Divide Ω into nonoverlapping rectangles $[k_i, k_{i+1}] \times [z_j, z_{j+1}]$, where k_i is the ith grid point for capital and z_j is jth grid point for the technology shock.
- Clearly $\Omega = \cup_{i,j} [k_i, k_{i+1}] \times [z_j, z_{j+1}]$.

Our Functional Basis

- Set $I_{fe}(k,z;\theta) = \sum_{i,j}\theta_{ij}\Psi_{ij}(k,z) = \sum_{i,j}\theta_{ij}\widehat{\Psi}_i(k)\widetilde{\Psi}_j(z)$ where

$$\widehat{\Psi}_i(k) = \begin{cases} \frac{k-k_{i-1}}{k_i-k_{i-1}} & \text{if } k \in [k_{i-1}, k_i] \\ \frac{k_{i+1}-k}{k_{i+1}-k_i} & \text{if } k \in [k_i, k_{i+1}] \\ 0 \text{ elsewhere} \end{cases}$$

$$\widetilde{\Psi}_j(z) = \begin{cases} \frac{z-z_{j-1}}{z_j-z_{j-1}} & \text{if } z \in [z_{j-1}, z_j] \\ \frac{z_{j+1}-z}{z_{j+1}-z_j} & \text{if } z \in [z_j, z_{j+1}] \\ 0 \text{ elsewhere} \end{cases}$$

- Note that:
  1. $\Psi_{ij}(k,z) = 0$ if $(k,z) \notin [k_{i-1}, k_i] \times [z_{j-1}, z_j] \cup [k_i, k_{i+1}] \times [z_j, z_{j+1}]$ $\forall i, j$, i.e. the function is 0 everywhere except inside two elements.
  2. $I_{fe}(k_i, z_j; \theta) = \theta_{ij}$ $\forall i, j$, i.e. the values of θ specify the values of c_{fe} at the corners of each subinterval $[k_i, k_{i+1}] \times [z_j, z_{j+1}]$.

Residual Function I

- Define $U_c\left(k_{t+1}, z_{t+1}\right)_{fe}$ as the marginal utility of consumption evaluated at the finite element approximation values of consumption and leisure.
- From the Euler equation we have a residual equation:

$$R\left(k_t, z_t; \theta\right) =$$
$$\frac{\beta}{\pi^{0.5}} \int_{-1}^{1} \left[\frac{U_c\left(k_{t+1}, z_{t+1}\right)_{fe}}{U_c\left(k_{t+1}, z_{t+1}\right)_{fe}} r\left(k_{t+1}, z_{t+1}\right) \right] \exp\left(-v_{t+1}^2\right) dv_{t+1} - 1$$

- A Galerkin scheme implies that we weight the residual function by the basis functions and solve the system of θ equations

$$\int_{[0,k_{\max}]\times[-1,1]} \Psi_{i,j}\left(k, z\right) R\left(k, z; \theta\right) dzdk = 0 \quad \forall i, j$$

on the θ unknowns.

Residual Function II

- Since $\Psi_{ij}(k,z) = 0$ if $(k,z) \notin [k_{i-1}, k_i] \times [z_{j-1}, z_j] \cup [k_i, k_{i+1}] \times [z_j, z_{j+1}]$ $\forall i,j$ we have

$$\int_{[k_{i-1},k_i]\times[z_{j-1},z_j]\cup[k_i,k_{i+1}]\times[z_j,z_{j+1}]} \Psi_{i,j}(k,z)\, R(k,z;\theta) dzdk = 0 \quad \forall i,j$$

- We use Gauss-Hermite for the integral in the residual equation and Gauss-Legendre for the integrals in Euler equation.

- We use 71 unequal elements in the capital dimension and 31 on the λ axis. To solve the associated system of 2201 nonlinear equations we use a Quasi-Newton algorithm.