Computing and Estimating RBC Models With Global Methods

David Wiczer

UMN Economics

October 13, 2010

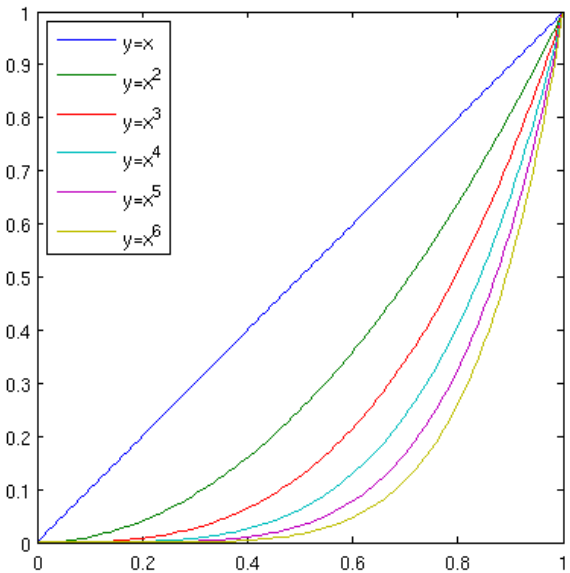
Outline

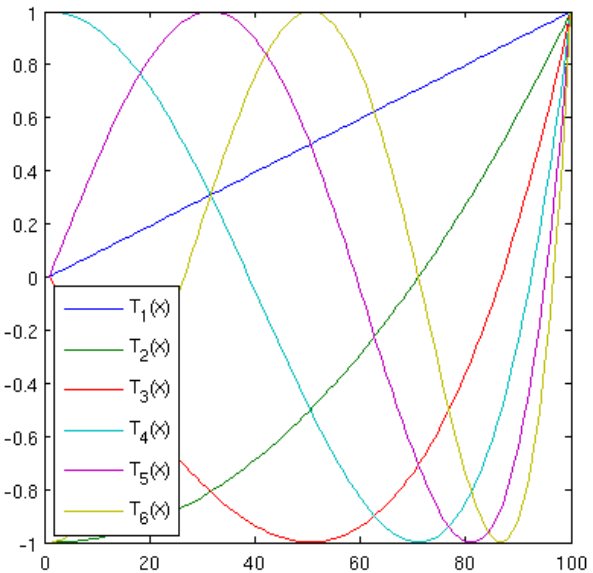
Function representation by orthogonal functions

- We make the approximation that $f(x) \approx \sum_{i=0}^{I} \alpha_i \phi_i(x)$ where ϕ are known functions.
- To "store" $f(\cdot)$, we really want to store the vector $\{\alpha_i\}$
- We need to choose good polynomials to economize on the number of functions we need to evaluate each time we approximate $f(x)$.

Chebyshev functions

Polynomials become too similar. Chebyshev functions are orthogonal, meaning accuracy gains do not diminish as we increase the number of functions.





These follow the recursive defintion:

$T_0(x) = 1,\ T_1(x) = x,\ T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$ or

$T_i(x) = \cos(i\cos^{-1}(x))$

Where to evaluate

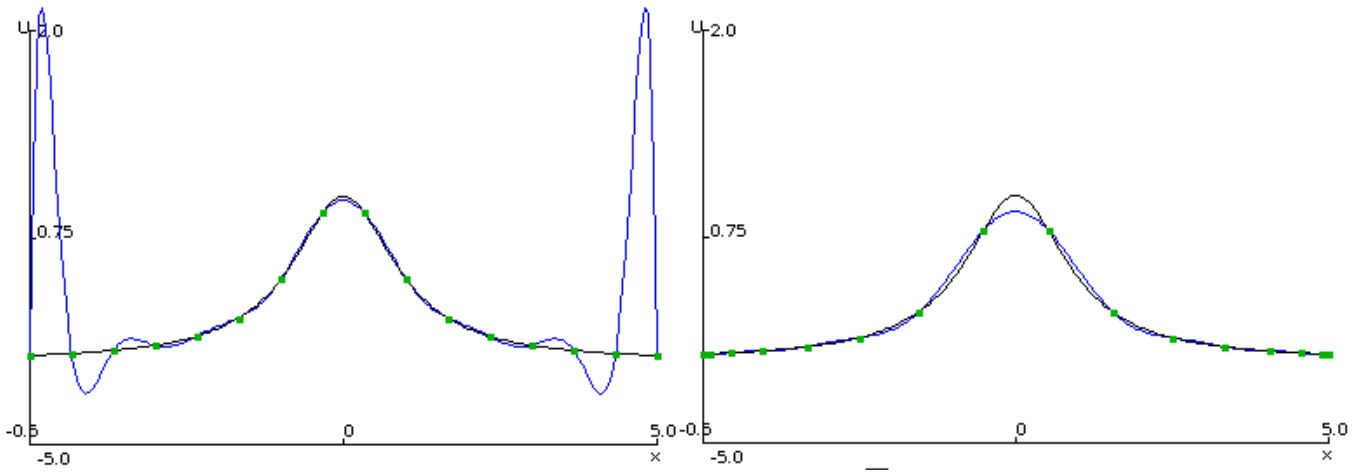
- The n^{th} order Chebyshev function has roots given by $z_i = -\cos\left(\frac{(2i-1)\pi}{2n}\right)$
- If you approximate with n^{th} order function, you need to fit at $m \geq n+1$ points.
- These functions are only defined on $[-1,1]$, so you need to transform your domain. The best way is to ensure that bounds of your domain fall on the first and last zero
- To transform from $\{x_i\} \in [\underline{x}, \bar{x}]$ with $\{z_i\} \in (-1,1)$ use

$$x(z) = \frac{1}{2}(sec(\frac{\pi}{2m})z + 1)(\bar{x} - \underline{x}) + \underline{x}$$

- You can invert that for $z(x)$

Perils of ignoring the math

- Notice, the representation always oscilates around the true. This is the “equioscilation principle”



- Evenly spaced grids make for terrible approximation. Even with other base functions, the Chebyshev zeros are often good (says Judd)
- Why global functions instead of piecwise? To increase the approximation accuracy *everywhere* we can easily add points/bases.

Defining weights

- With $m = n + 1$, "collocation," weights to represent f are

$$\alpha_j = \frac{2}{m} \sum_{i=1}^{m} T_j(z_i) f(z_i)$$

- The 0 order, $\alpha_0 = \frac{1}{m} \sum_{i=1}^{m} f(z_i)$
- The weights will be decreasing in absolute value. We can "economize" on the number of functions to evalueate by cutting off at some earlier n where θ_n is small.

Algorithm for solving a growth model

Our plan is to approximate the value function by Chebyshev functions. Why is this a bad idea?

1. We consider capital $k \in [\underline{k}, \bar{k}]$, which means we'll evaluate at $\{k_i\}$ corresponding to $\{z_i\}$
2. For the initial guess $V^0(k)$ find $\{\alpha_j^0\}$ such that $V^0(k) \approx \sum_{j=0}^{n} \alpha_j^0 T_j(z(k))$ by solving for $\alpha_j^0 = \frac{2}{m} \sum_{i=1}^{m} T_j(z_i) f(z(k_i))$
3. At each k_i solve $V^1(k_i) = \max_{k'} u(k_i, k') + \beta \sum_{j=0}^{n} \alpha_j^0 T_j(z(k'))$
4. Update by solving for $\{\alpha_j^1\} \ : \ V^1(k) \approx \sum_{j=0}^{n} \alpha_j^1 T_j(z(k))$
5. Stop when the decision rule converges given $\{\alpha^t\}, \{\alpha^{t+1}\}$

Outline

The idea

- I can compute the model, but the direct mapping to the data is not obvious.
- Instead, I choose another model which will represent the statistical properties of the data. For the “auxilliary model” I can (pick one)
  - Write moment condtions
  - Write a likelihood function
- Then I will get model parameters to minimize the distance to this statistical description of the data.

The method

- Find parameters such that estimating the auxilliary model on data and model output is the same.
- Consider finding "true model" parameters β, $\hat{\theta}^s(\beta) = \arg\max_\theta \mathcal{L}(\{x_s(\beta)\}; \theta)$
- $\mathcal{L}$ is this other statistical description of the data

The method

- From the data, find $\hat{\theta}^d = \arg\max_\theta \mathcal{L}(\{x\}; theta)$
- Choose β such that $\hat{\beta} = \arg\max_\beta \mathcal{L}(\{x\}; \hat{\theta}^s(\beta))$.
- There is an unknown function $\tilde{\theta}(\beta)$ that "binds" the paremeters. Find it by simulating and averaging $\hat{\theta}^s(\beta)$
- In practice, we minimize the distance $(\hat{\theta}^d - \tilde{\theta}(\beta))^T W (\hat{\theta}^d - \tilde{\theta}(\beta))$

The method

- To summarize, This involves 2 steps
- We have decisions of the form $\sum \phi(x, \alpha; \beta, \theta)$
- I solve the model around β, θ for α
- I then solve for $\hat{\theta}^s$ that is optimal in terms of my auxilliary model
- Victor's suggestion: I don't want to be solving for α too accurately when $\hat{\theta}^s$ is far from $\hat{\theta}^d$
- Can I jointly solve for α, β, θ?

An example

Following Smith (1993), we take a simple RBC model:

$$\begin{aligned}
\max \quad & \sum_{t=0}^{\infty} \beta^t u(c_t, l_t) \\
c_t + x_t &= z_t k_t^{\alpha} l_t^{1-\alpha} \\
k_{t+1} &= (1-\delta) k_t + b_t x_t \\
\log z_t &= \rho \log z_{t-1} + \epsilon_t \ : \ \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \\
\log b_t &= \phi \log b_{t-1} + \upsilon_t \ : \ \upsilon \sim \mathcal{N}(0, \sigma_\upsilon^2)
\end{aligned}$$

Decision rules $x(k, z, b)$ and $h(k, z, b)$ are complicated functions of the parameters. We cannot (generally) write an explicit likelihood function.

The appropriate auxilliary model

- We know that this model is well approximated by a linearized form of states and decision rules
- Then the decision rules are linear in the endogenous and exogenous states

$$\begin{aligned} k_{t+1} &= \Omega k_t + \Gamma u_t \\ l_{t+1} &= P k_t + \Upsilon u_t \end{aligned}$$

- With obbservations of k_t, l_t, we could obviously estimate $\Omega, \Gamma, P, \Upsilon$ by ML, but these are tricky functions of the true structural parameters.
- It is not so different from a VAR, Smith (1993) uses

$$s_{t+1} = [y_{t+1}\ x_{t+1}\ y_t\ x_t]^T = Q s_t + E_t$$

- I'll use $s_t = [k_t\ l_t\ y_t\ (w_t l_t)]^T$

Execution

1. Estimate the auxilliary model with the data.
2. Draw your random numbers first. Then the model moves around these numbers. We are doing a path simulation, so you're drawing $\{\epsilon_t, \upsilon_t\}$. You need to do H simulations length T
3. Solve the model for decision rules as functions of the draws. Estimate the auxilliary model H times.

Execution

1. Estimate the auxilliary model with the data.
2. Draw your random numbers first. Then the model moves around these numbers. We are doing a path simulation, so you're drawing $\{\epsilon_t, \upsilon_t\}$. You need to do H simulations length T
3. Solve the model for decision rules as functions of the draws. Estimate the auxilliary model H times.
4. Minimize the distance

$$(\hat{\theta}^d - \frac{1}{H}\sum_{h=1}^{H}\hat{\theta}_h(\beta))^T V(\beta)^{-1}(\hat{\theta}^d - \frac{1}{H}\sum_{h=1}^{H}\hat{\theta}_h(\beta))$$

5. For the weight matrix, use the inverse of the variance at each estimate

Variance estimates

- The most obvious is to use the Delta method:
- The derivative of the mapping $\theta^s(\beta)$ can be evaluated $J = \frac{\partial \theta^s(\beta)}{\partial \beta}$
- The variance of the auxilliary model comes from our intermediate estimate $E\ E^T$
- Then $V(\beta) = JE\ E^T J^T$
- This is particularly computationally demanding, so there are analytical forms, see Gourierioux, Monfort and Renaux (1993)

The result

Smith uses US data from 1947:Q1 to 1988:Q4, I'm going to use 1964:Q1 to 2009:Q4

	$\hat{\alpha}$	$\hat{\rho}$	$\hat{\sigma}_\epsilon$	$\hat{\phi}$	$\hat{\sigma}_\upsilon$
Smith	0.2415	0.8961	0.01023	0.7851	0.0524
	(0.3798)	(0.0849)	(0.00078)	(0.0689)	(0.00889)
Me	0.347	0.928	0.0047	0.6988	0.0123
	(0.3798)	(0.0849)	(0.00078)	(0.0689)	(0.00889)

You can get my code from my website:
http://www.econ.umn.edu/∼wicze006/computation.html

Outline

Representing a value function with an exogenous state

- Basically we will discretize a dimension of $V(k, e)$.
- Choose $\{e_s\}_{s=1}^{S}$ to evaluate and solve for $\{\theta_j^{e_s}\} \ : \ V(k, e_s) \approx \sum_{j=0}^{n} \theta_j^{e_s} T_j(z(k))$
- Then the value function, evaluated at chebyshev nodes $\{k_i\}$ and exogenous state $\{e_s\}$ is defined by

$$V(k_i, e_s) = \max_{k'} u(k_i, k', e_s) + \beta \sum_{r=1}^{S} \pi(e_r|e_s) \sum_{j=0}^{n} \theta_j^r T_j(z(k'))$$

Representing a value function with an exogenous state

- We don't worry about interpolating in the exogenous direction. We never need values in between or the derivative because we never choose an e'.
- For a second endogenous state variable, linear interpolation between smooth Chebyshev approximations is not a good strategy. Use bicubic splines or other multidimensional basis functions. (code for this is also on my website)

Discretizing a process

- The general idea is to take a distribution f and find a set of points $\{x_i\}$ and assign to them probability $Pr[x_i] = \int_{c_i}^{c_{i+1}} f(x)dx$ for some set of cutoff points $\{c_i\} \ : \ c_i \in (x_i, x_{i+1})$
- Generally we are concerned with making a markov chain and generally dynamic process will not be row-identical.
- With Tauchen (1986), you can conveniently pick the points of the process at which to evaluate and choose probabilities to suit.
- To accurately approximate a process with the fewest points, we can often do better. However, that may not be the point

Tauchen's Method

- Fit $z' = \rho z + \epsilon \ : \ \epsilon \sim \mathcal{N}(0, \sigma^2)$ with $n \times n$ transition matrix Π and vector of points $\{z_i\}$
- The stationary distribution of $z \sim \mathcal{N}(0, \sigma_z^2) \ : \ \sigma_z^2 = \frac{\sigma^2}{1-\rho^2}$
- Choose a grid for $\{z_i\}$, commonly equidistant cutoffs between $z_0 = \lambda\sigma_z, \ z_I = \lambda\sigma_z$ and let $c_i = \frac{z_{i+1}+z_i}{2}$
- Because
  $Pr[z' \in [c_{j-1}, c_j]|z_i] = Pr[\epsilon' \in (c_j - \rho z_i, c_{j-1} - \rho z_i)|z_i]$

$$p_{ij} = \Phi\left(\frac{c_j - \rho z_i}{\sigma}\right) - \Phi\left(\frac{c_{j-1} - \rho z_i}{\sigma}\right)$$

- on the ends: $p_{i0} = \Phi\left(\frac{c_0 - \rho z_i}{\sigma}\right) \quad p_{iI} = 1 - \Phi\left(\frac{c_{I-1} - \rho z_i}{\sigma}\right)$

Rouwenhorst method

- With extremely persistent processes, we often need many points in Tauchen and the preformance is poor.
- From Kopecky and Suen (2009) a simple way to construct the matrix. The size-N grid is evenly spaced over the domain.
- The basic idea is to approximate a normal distribution with binomials. CLT says this works

Rouwenhorst method

- Let $\Phi(t,i) = [p + (1-p)t]^{N-i}(1-p+qt)^{i-1}$
- For $i = 1, \ldots, N$ this yields $\Phi(t,i) = \sum_{j=1}^{N} \pi_{i,j} t^{j-1}$ and $\Pi_N = [\pi_{ij}]$
- This is equivalent to a (more intuitive) matrix-recursive formulation from the original paper. For $N = 2$

$$\Pi_2 = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}$$

$$\begin{aligned} \Pi_{N\geq 3} =& p \begin{pmatrix} \Pi_{N-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} + (1-p) \begin{pmatrix} \mathbf{0} & \Pi_{N-1} \\ 0 & \mathbf{0}^T \end{pmatrix} \\ &+ (1-q) \begin{pmatrix} \mathbf{0}^T & 0 \\ \Pi_{N-1} & \mathbf{0} \end{pmatrix} + q \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \Pi_{N-1} \end{pmatrix} \end{aligned}$$

- This has the flavor of using binomials to approximate a normal.

Rouwenhorst method

- We can choose p, q to hit various moments of a distribtution, which they outline in the paper.
- Because z is normal, there are only so many targets. Let $p = q = \frac{1+\rho}{2}$ and spacing is $\sqrt{N-1}\sigma$

Table 1: Selected Moments of the Markov Chain

Conditional Mean	$\mathrm{E}(y_{t+1}\|y_t = \overline{y}_i)$	$(q-p)\,\psi + (p+q-1)\,\overline{y}_i$
Conditional Variance	$\mathrm{var}(y_{t+1}\|y_t = \overline{y}_i)$	$\frac{4\psi^2}{(N-1)^2}\left[(N-i)\,(1-p)\,p + (i-1)\,q\,(1-q)\right]$
Unconditional Mean	$\mathrm{E}(y_t)$	$\frac{(q-p)\psi}{2-(p+q)}$
Unconditional Second Moment	$\mathrm{E}(y_t^2)$	$\psi^2\left\{1 - 4s\,(1-s) + \frac{4s(1-s)}{N-1}\right\}$
First-order Autocovariance	$\mathrm{Cov}(y_t, y_{t+1})$	$(p+q-1)\mathrm{var}(y_t)$
First-order Autocorrelation	$\mathrm{Corr}(y_t, y_{t+1})$	$p+q-1$