

Projection Methods

David N. DeJong
University of Pittsburgh

Spring 2008, Revised Spring 2010

Motivation

To date, we have developed tools for

- ▶ converting model environments into non-linear first-order systems of the form

$$\Gamma(E_t z_{t+1}, z_t, v_t) = 0, \quad (1)$$

- ▶ and approximating the solution to these systems (log-)linearly, in the form

$$\begin{aligned} x_{t+1} &= Fx_t + Gv_t \\ &= Fx_t + e_t, \end{aligned} \quad (2)$$

where, e.g., $x_{it} = \ln \frac{z_{it}}{z_i}$.

Motivation, cont.

For many purposes, the approximation error associated with linear approximations is unacceptable. Specifically:

- ▶ non-linearities associated with policy functions are often of direct interest to the question at hand (e.g., in making judgements regarding the importance of precautionary motives in driving consumption/savings decisions: Hugget and Ospina, 1993 *JME*, “Aggregate Precautionary Savings: When is the Third Derivative Irrelevant?”)
- ▶ desired measurements can be highly sensitive to model approximation errors (e.g., second-order model approximation errors accumulate into first-order approximation errors in associated likelihood functions: Fernandex-Villaverde and Rubio-Ramirez, 2005 *JAE*; 2007 *REStud*).

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$l\$ -Dimensional State Space](#)[Judging Approximation Quality](#)[Calculating Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Motivation, cont.

In such cases, we must obtain more accurate (non-linear) model approximations. There are (at least) three alternative methods for doing-so:

- ▶ value/policy function iterations
- ▶ higher-order Taylor Series approximations (i.e., perturbation methods; Judd and Guu, 1997 *JEDC*; Schmitt-Grohe and Uribe, 2004 *JEDC*)
- ▶ projection methods

In seeking non-linear approximations, we typically work with variables expressed in levels (detrended when appropriate).

Let s_t denote the vector of state variables contained in z_t , with law of motion determined by

$$s_t = f(s_{t-1}, v_t), \quad (3)$$

and let c_t denote the vector of control variables contained in z_t .

Notation, cont.

The solution to the model we seek is a policy function of the form

$$c_t = c(s_t). \quad (4)$$

The policy function is derived as the solution to

$$F(c(s)) = 0, \quad (5)$$

where $F()$ is an operator defined over function spaces.

Notation, cont.

For example, in the one-tree model

$$p_t = \beta e^{(1-\gamma)g} E_t \left[\left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} (d_{t+1} + p_{t+1}) \right] \quad (6)$$

$$c_t = d_t + q_t \quad (7)$$

$$d_t = \bar{d} e^{u_{dt}}, \quad u_{dt} = \rho_d u_{dt-1} + \varepsilon_{dt} \quad (8)$$

$$q_t = \bar{q} e^{u_{qt}}, \quad u_{qt} = \rho_q u_{qt-1} + \varepsilon_{qt}, \quad (9)$$

(8) and (9) jointly represent (3), (7) trivially represents the policy function for consumption, and the policy function for p_t emerges as the solution to the functional equation (6).

Overview

References: Ch. 10.2, Judd (1998), Ch. 6.

Projection methods involve the construction of approximated policy functions $\hat{c}(s)$ such that the associated approximation

$$F(\hat{c}(s)) = 0$$

is 'satisfactory'.

Specific methods differ along two dimensions:

- ▶ the functional form over s used to construct $\hat{c}(s)$;
- ▶ the criterion used to judge whether the approximation $F(\hat{c}(s)) = 0$ is satisfactory.

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$l\$ -Dimensional State Space](#)[Judging](#)[Approximation](#)[Quality](#)[Calculating](#)[Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing](#)[Ranges for](#)[Approximation](#)[Judging Quality of](#)[Fit](#)[Application](#)

Overview, cont.

Justification for this approach stems from the Weierstrass Theorem:

Any continuous function $c()$ may be approximated uniformly well over the range $s \in [\underline{s}, \bar{s}]$ by a sequence of r polynomials $p_r(s)$. That is,

$$\lim_{r \rightarrow \infty} \max_{s \in [\underline{s}, \bar{s}]} |c(s) - p_r(s)| = 0.$$

Examples

For now, let s be one-dimensional

Example 1: Linear combination of monomials in s ,
 $\{s^0, s^1, \dots, s^r\}$:

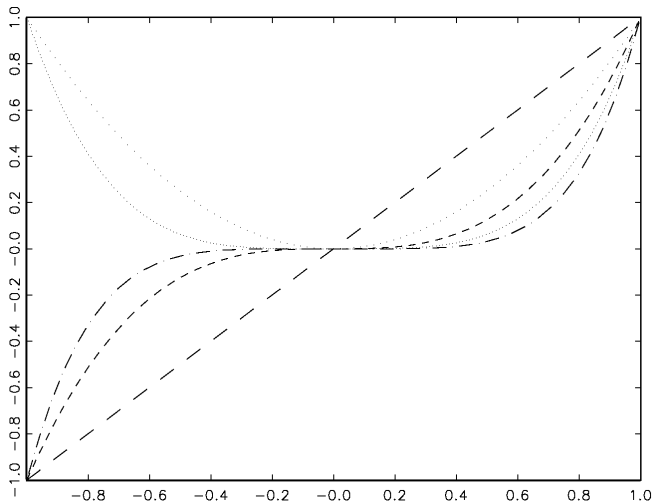
$$p_r(s) = \sum_{i=0}^r \chi_i s^i.$$

In using $p_r(s)$ to construct $\hat{c}(s, \chi)$, the goal is to choose the parameters collected in the vector χ to provide an optimal characterization of $c(s)$.

This turns out to provide poor approximations in general, due to similarities in the behavior of the individual elements of $p_r(s)$.

Examples, cont.

Behavior of $\{s^0, s^1, \dots, s^r\}$ over $[-1 \ 1]$:



Motivation

Notation

Overview

Examples of
Polynomials

**One-Dimensional
State Space**

r -Dimensional State
Space

Judging

Approximation
Quality

Calculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
Approximation

Judging Quality of
Fit

Application

Example 2: Tent Function

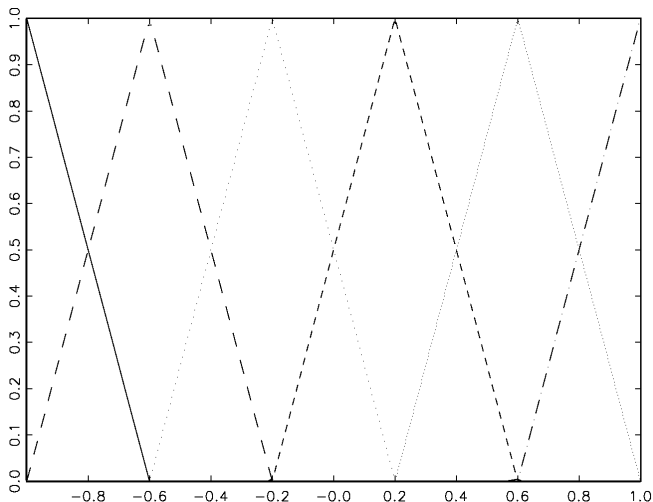
$$p_r(s) = \sum_{i=0}^r \chi_i p_r^i(s),$$

$$p_r^i(s) = \left\{ \begin{array}{ll} \frac{s-s^{i-1}}{s^i-s^{i-1}}, & s \in [s^{i-1}, s^i] \\ \frac{s^{i+1}-s}{s^{i+1}-s^i}, & s \in [s^i, s^{i+1}] \\ 0, & \text{otherwise.} \end{array} \right\}$$

Approximation methods involving tent functions are known as finite element methods.

Examples, cont.

Behavior of tent function over $[-1 \ 1]$:



Motivation

Notation

Overview

Examples of
Polynomials

**One-Dimensional
State Space**

l -Dimensional State
Space

Judging

Approximation
Quality

Calculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
Approximation

Judging Quality of
Fit

Application

Examples, cont.

Example 3: Chebyshev polynomial.

Background: The Chebyshev polynomial is an example of an orthogonal polynomial.

The definition of an orthogonal polynomial is based on the inner product between two functions f_1 and f_2 , given the weighting function w :

$$\langle f_1, f_2 \rangle = \int_{\underline{s}}^{\bar{s}} f_1(s) f_2(s) w(s) ds.$$

The family of polynomials $\{\varphi_r\}$ is defined to be mutually orthogonal with respect to $w(s)$ if and only if $\langle \varphi_r, \varphi_q \rangle = 0$ for $r \neq q$.

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$\ell\$ -Dimensional State Space](#)[Judging](#)[Approximation Quality](#)[Calculating Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Examples, cont.

The Chebyshev polynomial is defined over $s \in [-1, 1]$; its associated the weighting function is

$$w(s) = (1 - s^2)^{-1/2}.$$

It is given by

$$p_r(s) = \sum_{i=0}^r \chi_i T_i(s),$$

where

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

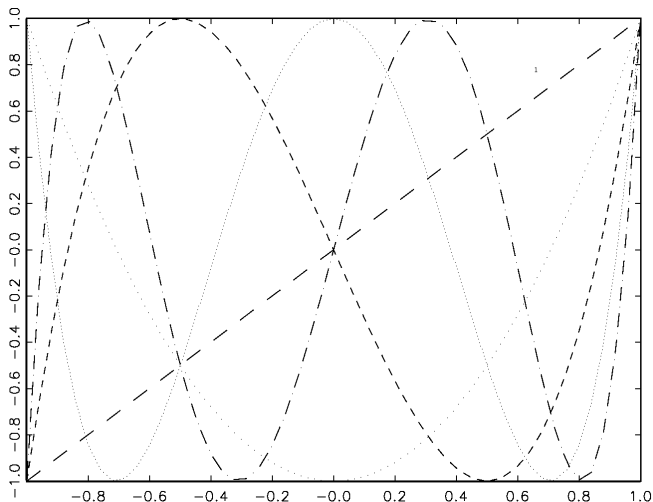
Examples, cont.

For a given value of s , $p_r(s)$ may be constructed as follows (Judd, 1998):

- ▶ Set $T_0(s) = 1$
- ▶ Set $T_1(s) = s$.
- ▶ Perform the recursion $T_{i+1}(s) = 2sT_i(s) - T_{i-1}(s)$,
 $i = 3, \dots, r$.
- ▶ Collecting the $T_i(s)$ terms in the $r + 1 \times 1$ vector $T(s)$,
calculate $p_r(s) = T(s)' \chi$.

Examples, cont.

Behavior of $T_i(s)$ over $[-1 \ 1]$:



Motivation

Notation

Overview

Examples of
Polynomials

**One-Dimensional
State Space**

l -Dimensional State
Space

Judging

Approximation
Quality

Calculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
Approximation

Judging Quality of
Fit

Application

Examples, cont.

Note that while s must be constrained between $[-1, 1]$ in working with Chebyshev polynomials, a simple transformation may be used to map a state variable defined over a general range $[\underline{s}, \bar{s}]$ into a variable defined over the range $[-1, 1]$.

For example, for an element of s ranging $\pm\omega s^*$ units above and below the steady state value s^* the transformation

$$\tilde{s} = \frac{s - s^*}{\omega s^*}$$

yields the desired range.

ℓ -Dimensional State Space

Now consider $s = (s_1 \ s_2 \ \dots \ s_\ell)'$.

Two leading approaches to approximation in this case involve the use of *tensor products* and *complete polynomials*.

Let $p_{r_j}(s_j)$ denote an r_j^{th} -order sequence of polynomials specified for the j^{th} element of s . Then the ℓ -dimensional tensor product of $p_{r_j}(s_j)$, $j = 1, \dots, \ell$, is given by

$$P = \prod_{j=1}^{\ell} p_{r_j}(s_j).$$

ℓ -Dimensional State Space, cont.

Motivation

Notation

Overview

Examples of
PolynomialsOne-Dimensional
State Space ℓ -Dimensional State
SpaceJudging
Approximation
QualityCalculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
ApproximationJudging Quality of
Fit

Application

For example, with $\ell = 2$, $r_1 = 2$, $r_2 = 3$, and $p_{r_j}(s_j)$ representing Chebyshev polynomials, we have

$$P = (1 + s_1 + T_2(s_1)) (1 + s_2 + T_2(s_2) + T_3(s_2)),$$

where $T_i(s_j)$ is the i^{th} -order term corresponding with the j^{th} element of s

I-Dimensional State Space, cont.

Using a given tensor product, the approximation $\widehat{c}(s, \chi)$ is constructed using

$$\widehat{c}(s, \chi) = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_\ell=1}^{r_\ell} \chi_{i_1 i_2 \dots i_\ell} P_{i_1 i_2 \dots i_\ell}(s_1, s_2, \dots, s_\ell),$$

where

$$P_{i_1 i_2 \dots i_\ell}(s_1, s_2, \dots, s_\ell) = p_{i_1}(s_1) p_{i_2}(s_2) \dots p_{i_\ell}(s_\ell).$$

ℓ -Dimensional State Space, cont.

In the example above, the elements $P_{i_1 i_2 \dots i_\ell}(s_1, s_2, \dots, s_\ell)$ can be constructed by forming the vectors

$$\begin{aligned}T^1 &= (1 \ s_1 \ T_2(s_1))', \\T^2 &= (1 \ s_2 \ T_2(s_2) \ T_3(s_2))',\end{aligned}$$

and retrieving the elements of the 3×4 matrix $T = T^1 T^{2'}$.
Moving to ℓ dimensions, construction may be achieved via recursion on

$$\text{vec}(T^i T^{i+1'}) T^{i+2}, \quad i = 1, \dots, \ell - 2.$$

ℓ -Dimensional State Space, cont.

An issue with the use of tensor products: as ℓ increases, the number of elements $\chi_{i_1 i_2 \dots i_\ell}$ that must be estimated to construct the approximation $\widehat{c}(s, \chi)$ increases exponentially. This is a manifestation of the curse of dimensionality.

One possible remedy: *complete polynomials*. For a polynomial of degree k , this refers to the collection of terms that appear in the k^{th} -order Taylor Series approximation of $c(s)$ about s^0 .

l -Dimensional State Space, cont.

For $k = 1$, the complete set of polynomials for the l -dimensional case is given by

$$P_1^l = \{1, s_1, s_2, \dots, s_l\}.$$

For $k = 2$ the set expands to

$$P_2^l = P_1^l \cup \{s_1^2, s_2^2, \dots, s_l^2, s_1 s_2, s_1 s_3, \dots, s_1 s_l, s_2 s_3, \dots, s_{l-1} s_l\},$$

etc.

In the two-dimensional case, while the tensor product of third-order polynomials involves 16 terms, the complete set of third-degree polynomials involves only 10 terms.

Judging Approximation Quality

Issue: for a given $\hat{c}(s, \chi) = p_r(s)$, what criterion should be used to select χ , and to judge the quality of the approximation this selection provides?

Specifically, in seeking to achieve

$$F(\hat{c}(s_i, \chi)) = 0,$$

the approximation we seek is defined by the parameter vector χ that minimizes

$$\langle F(\hat{c}(s, \chi)), f(s) \rangle = \int_{\underline{s}}^{\bar{s}} F(\hat{c}(s, \chi)) f(s) w(s) ds.$$

Alternative specifications for $f(s)$ and $w(s)$ differentiate projection methods along this second dimension.

[Motivation](#)
[Notation](#)
[Overview](#)
[Examples of Polynomials](#)
[One-Dimensional State Space](#)
 [\$l\$ -Dimensional State Space](#)
[Judging Approximation Quality](#)
[Calculating Expectations](#)
[Markov Processes](#)
[Quadrature Methods](#)
[Monte Carlo Integration](#)
[Starting Values](#)
[Establishing Ranges for Approximation](#)
[Judging Quality of Fit](#)
[Application](#)

Judging Approximation Quality, cont.

Three leading specifications;

- ▶ Weighted least squares (WLS)
- ▶ Galerkin method
- ▶ Collocation Method

WLS: $f(s) = F(\hat{c}(s, \chi))$, yielding the single-valued objective function

$$\langle F(\hat{c}(s, \chi)), f(s) \rangle = \sum_{i=1}^N F(\hat{c}(s_i, \chi))^2 w(s_i).$$

In this case, the optimal χ can be obtained via a numerical optimization procedure (e.g., Gauss's optimum).

Judging Approximation Quality, cont.

Galerkin: Finite element method (e.g., $\hat{c}(s, \chi)$ is constructed via a tent function), with $w(s) = 1$, and $f(s)$ is the sequence of basis functions $p_r^i(s)$, $i = 1, \dots, r$ used to construct $\hat{c}(s, \chi)$.

Here, the objective function is a system of r equations, which are to be solved by choice of the r -dimensional vector of coefficients χ :

$$\langle F(\hat{c}(s, \chi)), p_r^i(s) \rangle = \int_{\underline{s}}^{\bar{s}} F(\hat{c}(s, \chi)) p_r^i(s) ds = 0, \quad i = 1, \dots, r.$$

The integrals may be approximated using a sum over a range of distinct values chosen for s . Derivative-based methods are available for solving non-linear systems of this form (e.g., Gauss's nlsys).

[Motivation](#)
[Notation](#)
[Overview](#)
[Examples of Polynomials](#)
[One-Dimensional State Space](#)
 [\$r\$ -Dimensional State Space](#)
[Judging Approximation Quality](#)
[Calculating Expectations](#)
[Markov Processes](#)
[Quadrature Methods](#)
[Monte Carlo Integration](#)
[Starting Values](#)
[Establishing Ranges for Approximation](#)
[Judging Quality of Fit](#)
[Application](#)

Judging Approximation Quality, cont.

Motivation

Notation

Overview

Examples of
PolynomialsOne-Dimensional
State Space r -Dimensional State
SpaceJudging
Approximation
QualityCalculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
ApproximationJudging Quality of
Fit

Application

Collocation: Set $w(s) = 1$, and specify $f(s)$ as the indicator function $\delta(s - s_i)$, $i = 1, \dots, r$, with

$$\delta(s - s_i) = \begin{cases} 1, & s = s_i \\ 0, & s \neq s_i \end{cases}.$$

Under $\delta(s - s_i)$, the functional equation $F(\cdot) = 0$ is restricted to hold exactly at r fixed points.

Judging Approximation Quality, cont.

When using Chebyshev polynomials for constructing $\widehat{c}(s, \chi)$, the *Chebyshev Interpolation Theorem* provides a means of optimizing the r choices of s used to construct $\delta(s - s_i)$.

Letting $\{\widehat{s}_i\}_{i=1}^r$ denote the roots of the r^{th} -order component $T_r(s)$, if $F(\widehat{c}(\widehat{s}_i, \chi)) = 0$ for $i = 1, \dots, r$ and $F(\widehat{c}(s, \chi))$ is continuous, $F(\widehat{c}(s, \chi))$ will be close to 0 over the entire range $[-1, 1]$.

The r roots of $T_r(s)$ are given by

$$\widehat{s}_j = \cos\left(\frac{(2j-1)\pi}{2r}\right), \quad j = 1, 2, \dots, r.$$

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$r\$ -Dimensional State Space](#)[Judging Approximation Quality](#)[Calculating Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Calculating Expectations

Issue: The objective function

$$F(\hat{c}(s)) = 0$$

often contains an expectations operator. E.g.,

$$p_t = \beta e^{(1-\gamma)g} E_t \left[\left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} (d_{t+1} + p_{t+1}) \right].$$

Lacking knowledge of the properties of p_t , this expectations calculation cannot be performed analytically.

Calculating Expectations, cont.

Four general approaches to calculating expectations:

- ▶ Work with Markov processes for SPs
- ▶ Work with Markov chain approximations of continuous stochastic processes (Tauchen, 1986 *Econ. Letters*)
- ▶ Use quadrature methods to approximate expectations
- ▶ Monte Carlo simulation

Definition: A SP is said to have the *Markov property* if for all $k \geq 1$ and for all t ,

$$\Pr(x_{t+1} | x_t, x_{t-1}, \dots, x_{t-k}) = \Pr(x_{t+1} | x_t).$$

Definition: A *Markov chain* is defined by:

- ▶ A vector x with r unique values x_i , $i = 1, \dots, r$.
- ▶ A transition matrix P , with (i, j) th element

$$P_{ij} = \Pr(x_{t+1} = e_j | x_t = e_i).$$

- ▶ An initialization vector π_0 , with i th element

$$\pi_{0i} = \Pr(x_0 = e_i).$$

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$r\$ -Dimensional State Space](#)[Judging](#)[Approximation Quality](#)[Calculating Expectations](#)**Markov Processes**[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Calculating Expectations, cont.

For a one-dimensional state space, given $x_t = x_i$, the conditional expectation

$$E_t f(x_{t+1})$$

is given by the weighted average

$$\sum_{j=1}^r f(x_j) P_{ij}.$$

Given $x_t = [x_i^1 | x_{ii}^2 | \dots | x_{ii\dots i}^n]$, the n -dimensional case generalizes to

$$\sum_j \sum_k \dots \sum_z f(x_j^1 | x_k^2 | \dots | x_z^n) P_{ij}^1 P_{iik}^2 \dots P_{ii\dots iz}^n$$

Motivation

Notation

Overview

Examples of
PolynomialsOne-Dimensional
State Space l -Dimensional State
Space

Judging

Approximation
QualityCalculating
Expectations**Markov Processes**

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
ApproximationJudging Quality of
Fit

Application

Quadrature Methods

Quadrature methods comprise a wide class of numerical tools available for approximating specific examples of integrals of the form

$$\int_a^b f(x) dx.$$

Gaussian quadrature methods are an important subclass. For univariate cases, they take the form

$$\int_a^b f(x) dx \approx \sum_{i=1}^n w_i f(x_i),$$

where the nodes x_i and weights w_i are chosen so that if $f(\cdot)$ were a polynomial of degree $2n - 1$, then the approximation will be exact given the use of n nodes and weights.

Quadrature Methods, cont.

Gauss-Hermite nodes and weights are tailored specifically for cases in which integrals are of the form

$$\int_{-\infty}^{\infty} f(x) e^{-x^2} dx,$$

which arises naturally in working with normal random variables.

Quadrature Methods, cont.

Example. Consider a simplification of the one-tree model in which q is eliminated, and utility is linear:

$$\begin{aligned}p_t(d_t) &= \beta e^{(1-\gamma)g} E_t [d_{t+1} + p_{t+1}(d_{t+1})], \\ \ln d_{t+1} &= \rho \ln d_t + \varepsilon_{t+1}.\end{aligned}$$

Then

$$\begin{aligned}p_t(d_t) &= \beta e^{(1-\gamma)g} E_t \left[e^{\rho \ln d_t + \varepsilon_{t+1}} + p_{t+1}(d_{t+1}) \right] \\ &= \beta e^{(1-\gamma)g} \frac{1}{\sigma_\varepsilon \sqrt{2\pi}} \int_{-\infty}^{\infty} f(\varepsilon_{t+1} | d_t) e^{-\frac{\varepsilon_{t+1}^2}{2\sigma_\varepsilon^2}} d\varepsilon_{t+1}, \\ f(\varepsilon_{t+1} | d_t) &= e^{\rho \ln d_t + \varepsilon_{t+1}} + p_{t+1}(d_{t+1}).\end{aligned}$$

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$\ell\$ -Dimensional State Space](#)[Judging](#)[Approximation Quality](#)[Calculating Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Quadrature Methods, cont.

Defining

$$\begin{aligned}\varepsilon &= \sqrt{2}\sigma_\varepsilon x \\ &\equiv g(x),\end{aligned}$$

and applying the change-of-variables formula

$$\int_a^b f(\varepsilon) d\varepsilon = \int_{g^{-1}(a)}^{g^{-1}(b)} f(g(x))g'(x) dx,$$

we obtain

$$\begin{aligned}p(d_t) &= \beta e^{(1-\gamma)g} \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(\sqrt{2}\sigma_\varepsilon x | d_t) e^{-x^2} dx \\ &\approx \beta e^{(1-\gamma)g} \frac{1}{\sqrt{\pi}} \sum_{i=1}^n w_i f(\sigma_\varepsilon \sqrt{2}x_i).\end{aligned}$$

Monte Carlo Integration

MC Integration entails the approximation of integrals via simulation. Given the ability to obtain drawings of s_{t+1}^i from the known conditional pdf $p(s_{t+1}|\Omega_t)$, conditional expectations of the form

$$E_t f(s_{t+1}) = \int f(s_{t+1}) p(s_{t+1}|\Omega_t) ds_{t+1}$$

may be approximated as

$$E_t f(s_{t+1}) \approx \frac{1}{N} \sum_{i=1}^N f(s_{t+1}^i).$$

Monte Carlo Integration, cont.

In our example, we must simulate drawings $[d_{t+1} \quad q_{t+1}]'$ given $[d_t \quad q_t]'$.

This amounts to obtaining drawings $[\varepsilon_{dt+1} \quad \varepsilon_{qt+1}]'$ from a $N(0, \Sigma)$ distribution, and computing

$$d_{t+1} = \exp\left(\left(1 - \rho_d\right) \bar{d} + \rho_d \ln d_t + \varepsilon_{dt+1}\right),$$

$$q_{t+1} = \exp\left(\left(1 - \rho_q\right) \bar{q} + \rho_q \ln q_t + \varepsilon_{qt+1}\right).$$

Monte Carlo Integration, cont.

In GAUSS, drawings $[\varepsilon_{dt+1} \quad \varepsilon_{qt+1}]'$ may be obtained as follows:

- ▶ `sqrtsig=chol(sig)'`;
- ▶ `epsdraw = sqrtsig*rndn(2,1)`;

Starting Values

Issue: Achievement of fit can be sensitive to starting values χ_0 . Recommended solution: select χ_0 using linear approximation.

From the approximation

$$x_{t+1} = Fx_t + Gv_t,$$

the elements of F are elasticities.

Starting Values, cont.

For example, from our example, the two non-zero coefficients in the price equation are (σ_d, σ_q) .

In terms of an approximation in levels, these coefficients appear as

$$p \approx p^* + \frac{p^*}{d^*} \sigma_d (d - d^*) + \frac{p^*}{q^*} \sigma_q (q - q^*) \\ + \frac{1}{2} \left(\frac{p^*}{d^*} \sigma_d \right) \left(\frac{p^*}{q^*} \sigma_q \right) (d - d^*) (q - q^*).$$

Starting Values, cont.

Given the use of a tensor product representation, the corresponding approximation of $\hat{p}(d, q, \chi)$ we seek is of the form

$$\begin{aligned} \hat{p}(d, q, \chi) \approx & \chi_{11} + \chi_{12} \left(\frac{d - d^*}{\omega_d} \right) + \chi_{21} \left(\frac{q - q^*}{\omega_q} \right) \\ & + \chi_{22} \left(\frac{d - d^*}{\omega_d} \right) \left(\frac{q - q^*}{\omega_q} \right) + \dots \end{aligned}$$

Matching terms yields the suggested starting values

$$\begin{aligned} \chi_{11} &= p^*, \quad \chi_{12} = \sigma_d \omega_d \frac{p^*}{d^*}, \quad \chi_{21} = \sigma_q \omega_q \frac{p^*}{q^*}, \\ \chi_{22} &= \frac{1}{2} \left(\sigma_d \omega_d \frac{p^*}{d^*} \right) \left(\sigma_q \omega_q \frac{p^*}{q^*} \right). \end{aligned}$$

[Motivation](#)
[Notation](#)
[Overview](#)
[Examples of Polynomials](#)

- One-Dimensional State Space
- ℓ -Dimensional State Space

[Judging Approximation Quality](#)
[Calculating Expectations](#)

- Markov Processes
- Quadrature Methods
- Monte Carlo Integration

[Starting Values](#)
[Establishing Ranges for Approximation](#)
[Judging Quality of Fit](#)
[Application](#)

Establishing Ranges for Approximation

State variables are stochastic processes, so approximation ranges are best constructed on the basis of corresponding pdfs (e.g., centered at steady state values, and ranging $\pm x$ standard deviations around these values).

Information regarding pdfs is available once again from the log-linear approximation

$$x_t = Fx_{t-1} + e_t,$$

$$x_t = [\hat{x}_1, \hat{x}_2, \dots]' \quad \hat{x}_i = \ln \frac{x_i}{\bar{x}_i}.$$

Motivation

Notation

Overview

Examples of
PolynomialsOne-Dimensional
State Space l -Dimensional State
SpaceJudging
Approximation
QualityCalculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
ApproximationJudging Quality of
Fit

Application

Approximation Ranges, cont.

The unconditional VCV matrix of x solves

$$\begin{aligned}\Sigma_x &= Exx' \\ &= EF(xx')F' + Eee' \\ &= F\Sigma_x F' + Q,\end{aligned}$$

and is thus obtained from

$$\text{vec}(\Sigma_x) = (I - F \otimes F')^{-1} + Q.$$

Square roots of the diagonal elements of Σ_x yield $\sigma(\hat{x}_i)$.

Motivation

Notation

Overview

Examples of
Polynomials

One-Dimensional
State Space

l -Dimensional State
Space

Judging

Approximation
Quality

Calculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
Approximation

Judging Quality of
Fit

Application

Approximation Ranges, cont.

Motivation

Notation

Overview

Examples of
PolynomialsOne-Dimensional
State Space l -Dimensional State
SpaceJudging
Approximation
QualityCalculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

**Establishing
Ranges for
Approximation**Judging Quality of
Fit

Application

Then $\sigma(x_i)$ is obtained from $\sigma(\hat{x}_i)$ using

$$\Delta \hat{x}_i = \Delta \ln \frac{x_i}{\bar{x}_i} \approx \frac{\Delta x_i}{x_i},$$

which implies

$$\sigma(x_i) \approx \bar{x}_i \sigma(\hat{x}_i).$$

Judging Quality of Fit

Given a proposal for χ , fit can be judged simply by plotting

$$F(\hat{c}(s, \chi))$$

over a range chosen for s , and judging proximity to zero.

Returning to the functional equation

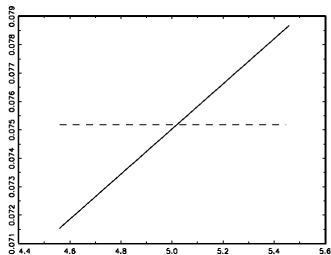
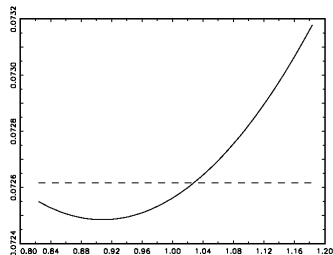
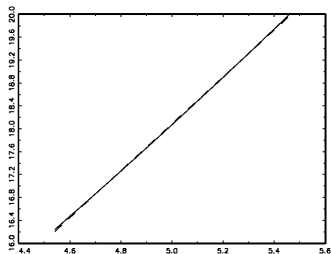
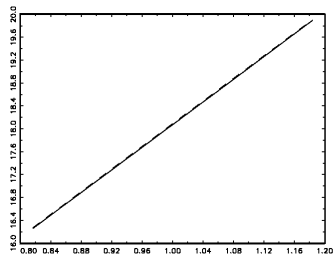
$$p_t = \beta e^{(1-\gamma)g} E_t \left[\left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} (d_{t+1} + p_{t+1}) \right],$$

approximate using a tensor product of Chebyshev polynomials over $[d_t \quad q_t]'$ (3rd and 4th order, respectively). Approximate E_t via MC simulation, with $N=10,000$. Use starting values χ_0 constructed from F as described above.

Parameterization: $\beta = 0.96$, $\gamma = 2$, $g = 0.013$, $\rho's = 0.9$, $\sigma_d = 0.02$, $\sigma_q = 0.01$, $corr(d, q) = 0.4$.

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$l\$ -Dimensional State Space](#)[Judging](#)[Approximation Quality](#)[Calculating Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Policy Functions and Slopes



Motivation

Notation

Overview

Examples of
PolynomialsOne-Dimensional
State Space l -Dimensional State
Space

Judging

Approximation
QualityCalculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

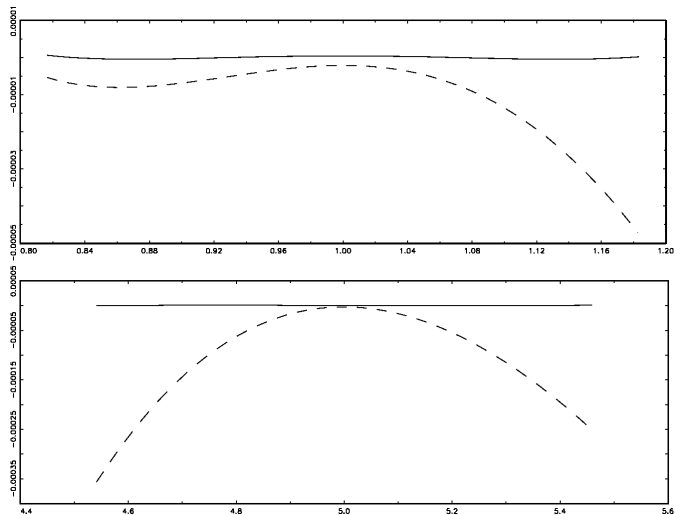
Starting Values

Establishing
Ranges for
ApproximationJudging Quality of
Fit

Application

Application, cont.

Fit



Proj. Methods

DND

Motivation

Notation

Overview

Examples of
Polynomials

One-Dimensional
State Space

l -Dimensional State
Space

Judging

Approximation
Quality

Calculating
Expectations

Markov Processes

Quadrature Methods

Monte Carlo
Integration

Starting Values

Establishing
Ranges for
Approximation

Judging Quality of
Fit

Application

Implications for Relative Standard Deviations, Nonlin Versus Lin

- ▶ Note: variables are measured in terms of logged deviations from steady state.
- ▶ For non-linear approximation, statistics obtained via simulation. Reference: Ch. 11.1, pp. 290-293.

σ_c/σ_p	0.537	0.544
σ_d/σ_p	1.101	1.074
σ_q/σ_p	0.527	0.537

Bottom line: differences are noticeable in this context, but their size is hard to interpret. Stay tuned for likelihood comparisons....

[Motivation](#)[Notation](#)[Overview](#)[Examples of Polynomials](#)[One-Dimensional State Space](#) [\$l\$ -Dimensional State Space](#)[Judging Approximation Quality](#)[Calculating Expectations](#)[Markov Processes](#)[Quadrature Methods](#)[Monte Carlo Integration](#)[Starting Values](#)[Establishing Ranges for Approximation](#)[Judging Quality of Fit](#)[Application](#)

Exercise: Construct a non-linear approximation of the optimal growth model; compare fit and relative standard deviations with those obtained using a log-linear approximation.

Details:

- ▶ Use a 4th-order Chebyshev polynomial for capital, 3rd-order for TFP shock
- ▶ Approximate expectations using Gauss-Hermite approximation
- ▶ Experiment with starting values for χ
- ▶ Experiment with alternative model parameterizations.