Reliably Computing Nonlinear Dynamic Stochastic Model Solutions: An Algorithm with Error Formulas

Gary S. Anderson*

Thursday 14th February, 2019: 7:29am

Abstract

This paper provides a technique for representing discrete time nonlinear dynamic stochastic time invariant maps. Using this new series representation, the paper augments a traditional solution strategy with an additional set of constraints thereby enhancing algorithm robustness and improving accuracy. In addition, it provides general formulae for evaluating the accuracy of proposed solutions. The technique can readily accommodate models with occasionally binding constraints and regime switching. The algorithm uses Smolyak polynomial function approximation in a way which makes it possible to exploit a high degree of parallelism.

*The analysis and conclusions set forth are those of the author and do not indicate concurrence by other members of the research staff or the Board of Governors. I would like to thank Luca Guerrieri, Christopher Gust, Hess Chung, Benjamin Johannsen and Robert Tetlow for their comments and suggestions. I would also like to thank Stephanie Harrington for her help with document preparation.
1 Introduction

This paper investigates the solution of discrete time dynamic stochastic infinite-horizon economies with bounded-time-invariant solutions. It describes how to obtain decision rules satisfying a system of first-order conditions, “Euler Equations”, and how to assess their accuracy. For an overview of techniques for solving nonlinear models, see (Judd, 1992; Christiano and Fisher, 2000; Doraszelskiy and Judd, 2004; Gaspar and Judd, 1997; Judd et al., 2014; Marcet and Lorenzoni, 1999; Judd et al., 2011; Maliar and Maliar, 2001; ?; ?).

This new series representation for bounded time series, adapted from (Anderson, 2010),

\[ x(x_{t-1}, \epsilon_t) = B x_{t-1} + \phi \psi \epsilon + (I - F)^{-1} \phi \psi \epsilon + \sum_{\nu=0}^{\infty} F^\nu \phi Z_{t+\nu}(x_{t-1}, \epsilon_t) \]

makes it possible to construct a series representation for any bounded time invariant discrete time map. As yet, the author has found no comparable use of a linear reference dynamical system for conveniently transforming one bounded infinite dimensional series into another.

It turns out that this representation provides a way to use Euler equation errors to approximate the magnitude of the approximation error in components of decision rules. This paper provides approximation explicitly relating the magnitude of components of the Euler equation errors to the magnitude of components of errors in the decision rule without resort to statistical inference or ancillary simulations.

In addition, the series representation provides exploitable constraints on model solutions that enhance algorithm reliability and improve solution accuracy. The series representation readily accommodates models even with multiple ergodic sets thereby providing a robust solution method for a wide class of models including models with occasionally binding constraints and/or regime switching.

The numerical implementation of the algorithm builds upon the work of (Judd et al., 2011; ?). In particular it uses the an-isotropic Smolyak Method, the adaptive parallelootope method (Judd et al., 2014) and precomputes all integrals required for the conditional expectations. The algorithm should scale well to large models as many of the algorithm’s components can be computed in parallel with limited amounts of information flowing between compute nodes.

The paper is organized as follows. Section 2 presents a useful new series representation for any bounded time series. Section 3 shows how to apply this series representation to time invariant maps. Section 4 provides formulae for approximating dynamic stochastic model error in proposed solutions. Section 5 presents a new solution algorithm for improving proposed solutions. Section 6 concludes.

1 This series representation should also prove useful for applications based on value function iteration but I defer treating this topic for future work.

2 Others have also studied approximation error for dynamic stochastic models (Judd et al., 2017; Santos and PeraltaÁÁAlva, 2005; Santos, 2000).
2 A New Series Representation For Bounded Time Series

Since numerous authors have developed solution algorithms and formulae for solving linear rational expectations models. It turns out that the solutions associated with these linear saddle point models, by quantifying the potential impact of anticipated future shocks, can play a new and somewhat surprising role in characterizing the solutions for highly non-linear models. This section describes the relationship between linear reference models and general bounded time series in route to discussing how these calculations can be useful for representing time invariant maps.

2.1 Linear Reference Models and a Formula for “Anticipated Shocks”

For any linear homogeneous \( L \) dimensional deterministic system that produces a unique stable solution,

\[
H\_{-1}x\_{t-1} + H_0x_t + H_1x_{t+1} = 0,
\]

inhomogeneous solutions

\[
H\_{-1}x\_{t-1} + H_0x_t + H_1x_{t+1} = \psi_{t}\epsilon + \psi_c
\]

can be computed as

\[
x_t = Bx_{t-1} + \phi\psi_{t}\epsilon + (I - F)^{-1}\phi\psi_c
\]

where

\[
\phi = (H_0 + H_1B)^{-1} \text{ and } F = -\phi H_1. \text{(Anderson, 2010)}
\]

It will be useful to collect the components of this solution for use later in the paper. In Section 2.2 I will use these matrices to construct a series representation for any bounded path. Define \( L \equiv \{H, \psi_{t\epsilon}, \psi_c; B, \phi, F\} \).

**Theorem 2.1** Consider any bounded discrete time path

\[
x_t \in \mathbb{R}^L \text{ with } \|x_t\|_{\infty} \leq \bar{X} \forall t > 0.
\]

Now, given the trajectories define \( z_t \) as

\[
z_t \equiv H\_{-1}x\_{t-1} + H_0x_t + H_1x_{t+1} - \psi_c - \psi_{t\epsilon_t}
\]

One can then express the \( x_t \) solving

\[
H\_{-1}x\_{t-1} + H_0x_t + H_1x_{t+1} = \psi_{t}\epsilon + \psi_c + z_t
\]

using

\[
x_t = Bx_{t-1} + \phi\psi_{t}\epsilon + (I - F)^{-1}\phi\psi_c + \sum_{\nu=0}^{\infty} F^\nu\phi z_{t+\nu}
\]

and

\[
x_{t+k+1} = Bx_{t+k} + (I - F)^{-1}\phi\psi_c + \sum_{\nu=0}^{\infty} F^\nu\phi z_{t+k+\nu+1} \forall t, k \geq 0.
\]
Consequently, given a bounded time series \(1\), and a stable linear homogeneous system, one can easily compute a corresponding series representation like \(2\) for \(x_t\). Interestingly, the linear model, \(H\), the constant term \(\psi_c\) and the impact of the stochastic shocks \(\psi_\epsilon\) can be chosen rather arbitrarily – the only constraints being the existence of a saddle-point solution for the linear system and that all \(z\)-values fall in the row space of \(H_1\). Further, note that since the eigenvalues of \(F\) are all less than one in magnitude, the formula supports the intuition that far distant shocks influence current conditions less than imminent shocks.

The transformation \(\{x_t, x_t, x_{t+1}, x_{t+2} \ldots\} \rightarrow L(x_{t-1}, \epsilon_t) \rightarrow \{z_t, z_{t+1}, z_{t+2}, \ldots\}\) is invertible. \(\{z_t, z_{t+1}, z_{t+2}, \ldots\} \rightarrow L(x_{t-1}, \epsilon_t)^{-1} \rightarrow \{x_t, x_t, x_{t+1}, x_{t+2} \ldots\}\) and the inverse can be interpreted as giving the impact of “fully anticipated future shocks” on the path of \(x_t\) in a linear perfect foresight model. Note that the reference model is deterministic, so that the \(z_t(x_{t-1}, \epsilon_t)\) functions will fully account for any stochastic aspects encountered in the models I analyze.

A key feature to note is that the series representation can accommodate arbitrarily complex time series trajectories, so long as these trajectories are bounded. Later, this observation will give us some confidence in the robustness of the algorithms constructing series representations for unknown families of functions satisfying complicated systems of dynamic non-linear equations.

### 2.2 Arbitrary Bounded Time Series and an “Almost” Arbitrary Linear Model

Consider the following matrix constructed from “almost” arbitrary coefficients

\[
\begin{bmatrix}
H_{-1} & H_0 & H_1
\end{bmatrix} = \begin{bmatrix}
0.1 & 0.5 & -0.5 & 1. & 0.4 & 0.9 & 1. & 1. & 0.9 \\
0.2 & 0.2 & -0.5 & 7. & 0.4 & 0.8 & 3. & 2. & 0.6 \\
0.1 & -0.25 & -1.5 & 2.1 & 0.47 & 1.9 & 2.1 & 2.1 & 3.9
\end{bmatrix}
\]

with \(\psi_c = \psi_\epsilon = 0\), \(\psi_z = I\). I have chosen these coefficients so that the linear model has a unique stable solution and, since \(H_1\) is full rank, its column space will span the column space for all non zero values for \(z_t\). The flexibility in choosing \(L\) will become more important in the context below where we must choose a linear reference model in a context where we may have little guidance about what might make a good choice. It turns out that the algorithm is very forgiving about the choice of \(L\). The solution for this system is

\[
B = \begin{bmatrix}
-0.0282384 & -0.0552487 & 0.00939369 \\
-0.0664679 & -0.700462 & -0.0718527 \\
-0.163638 & -1.39868 & 0.331726
\end{bmatrix}
\]

\[
\phi = \begin{bmatrix}
0.0210079 & 0.15727 & -0.0531634 \\
1.20712 & -0.0553003 & -0.431842 \\
2.58165 & -0.183521 & -0.578227
\end{bmatrix}
\]

\[
F = \begin{bmatrix}
-0.381174 & -0.223904 & 0.0946684 \\
-0.134352 & -0.189653 & 0.630956 \\
-0.816814 & -1.00633 & 0.0417094
\end{bmatrix}
\]

It is straightforward to apply \(2\) to the following three bounded families of time series paths

\[
x_{1,t} = \frac{\|x_{t-1}\|}{1 + \|\epsilon_t\|} D_\pi(t)
\]
where $D_\pi(t)$ gives the t-th digit of $\pi$

$$x_{2,t} = \frac{\|x_{t-1}\|}{(1 + \|\epsilon_t\|)^2} (-1)^t$$
$$ax_{3,t} = \epsilon_t$$

and the $\epsilon_t$ are a sequence of pseudo random draws from the uniform distributions $\mathcal{U}(-4, 4)$ produced subsequent to the selection of a random seed.

The upper three panels in Figure 1 display these three time series generated for a particular initial state vector and shock value. The first set of trajectories characterizes a function of the digits in the decimal representation of $\pi$ and was chosen to emphasize that continuity is not important for the existence of the series representation. The second set of trajectories oscillates between two values determined by a nonlinear function of the initial conditions, $x_{t-1}$ and the shock. The third set of trajectories characterizes a sequence of uniformly distributed random numbers based on a seed determined by a nonlinear function of the initial conditions and the shock. These paths were chosen to emphasize that the trajectories need not converge to a fixed point, and need not be produced by some linear rational expectations solution or by the iteration of a discrete-time map. The spanning condition and the boundedness of the paths are sufficient conditions for the existence of the series representation based on the linear reference model, $\mathcal{L}^2$.

The values for the $z$'s are not at all important and their relationship to the $x$ trajectories are rather unintuitive. What is important is that these values exist and are easy to compute. One can repeat the calculations for any given initial condition to produce a $z$ series exactly replicating the new set of trajectories. Thus, the family of $z$ functions along with equation 2 provide a series representation for the entire family of trajectories. Numerical linear algebra calculations can easily transform a $z_t$ series into an $x_t$ series and vice versa. This can be done for any path starting from particular initial conditions. Thus, any discrete time invariant map that generates families of bounded paths has a series representation like Equation 2.

$^3$Although potentially useful in some contexts, this paper will not investigate representations for families of unbounded, but slowly diverging trajectories.
2.3 Assessing $x_t$ Errors

The formula $2$ was derived in (Anderson, 2010) to compute the impact on the current state of fully anticipated future shocks. The formula characterizes the impact exactly. However one can contemplate the impact of at least two deviations from the exact calculation:

- One could truncate a series of correct values of $z_t$.
- One might have imprecise values of $z_t$ along the path.

2.3.1 Truncation Error

The series representation can compute the entire series to machine precision if all the terms are included, but, it will be useful to notice that the terms for state vectors closer to the initial time have the most important impact. One could consider approximating $\mathcal{X}_t$ by truncating the series at a finite number of terms.

$$\hat{x}_t \equiv Bx_{t-1} + \phi \psi \epsilon + (I - F)^{-1}\phi \psi c + \sum_{s=0}^{k} F^s \phi z_t$$

We can bound the series approximation truncation errors:

$$\sum_{s=k+1}^{\infty} F^s \phi \psi z = (I - F)^{-1}F^{k+1} \phi \psi z$$

$$\|x(x_{t-1}, \epsilon) - \hat{x}(x_{t-1}, \epsilon, k)\|_\infty \leq \left\| (I - F)^{-1}F^{k+1} \phi \psi z \right\|_\infty (\|H_{-1}\|_\infty + \|H_0\|_\infty + \|H_1\|_\infty) \mathcal{X}$$

Again, note that for approximating $x(x_{t-1}, \epsilon)$ the impact of a given realization along the path declines for those realizations which are more temporally distant. Figure 2 shows that this truncation error can be a very conservative measure of the accuracy of the truncated series. The orange line represents the computed approximation of the infinity norm of the difference between $x_t$ from the full series and a truncated series for different lengths of the truncation. The blue line shows the infinity norm of the actual difference between the $x_t$ computed using the full series and the value obtained using a truncated series. The series requires only the first 20 terms to compute the initial value of the state vector to high precision.

2.3.2 Path Error

We can assess the impact of perturbed values for $z_t$ by computing the maximum discrepancy required for the $z_t$ and applying the partial sum formula Thus, one can approximate $\mathcal{X}_t$ using

$$\hat{x}_t \equiv Bx_{t-1} + \phi \psi \epsilon + (I - F)^{-1}\phi \psi c + \sum_{s=0}^{\infty} F^s \phi (z_t + \Delta z_t)$$

Note yet again that for approximating $x(x_{t-1}, \epsilon)$ the impact of a given realization along the path declines for those realizations which are more temporally distant. However, we can
conservatively bound the series approximation errors by using the largest possible $\Delta z_t$ in the formula:

$$\|x(x_{t-1}, \epsilon) - \hat{x}(x_{t-1}, \epsilon, k)\|_\infty \leq \|(I - F)^{-1} \phi \psi_z\|_\infty \|\Delta z_t\|_\infty$$

3 Dynamic Stochastic Time Invariant Maps

Many dynamic stochastic models have solutions that fall in the class of bounded time invariant maps. Consequently, if we take care (see Section 3.2), we can apply the law of iterated expectations, to compute conditional expected solution paths forward from any initial value $x_{t-1}$ and realization of $\epsilon_t$. Subsequently, we can use the family of conditional expectations paths along with a contrived linear reference model to generate a series representation for the model solutions and to approximate errors.

The series representation will provide a linearly weighted sum of $z_t(x_{t-1}, \epsilon_t)$ functions that give us an approximation for the model solutions. So long as the trajectories are bounded, the time invariant maps can be represented using the framework from section 2. In this section I will begin with a familiar RBC model example.

Economists have long used similar manipulation of time series paths for dynamic models. Indeed, Potter constructs his generalized impulse response functions using differences in conditional expectations pathsPotter (2000); Koop et al. (1996).

Later, in order to handle models with regime switching and occasionally binding constraints, I will need to consider more complicated collections of equation systems with Boolean gates. I will show how to apply the series formulation and to approximate the errors for these models as well.
3.1 An RBC Example

I consider a model described in Maliar and Maliar (2005)\(^6\):

\[
\begin{align*}
\max & \left\{ u(c_t) + E_t \sum_{t=1}^{\infty} \delta^t u(c_{t+1}) \right\} \\
c_t + k_t = (1 - d)k_{t-1} + \theta_t f(k_t) \\
f(k_t) &= k_t^\alpha \\
u(c) &= \frac{c^{1-\eta} - 1}{1 - \eta}
\end{align*}
\]

The first order conditions for the model are

\[
\begin{align*}
\frac{1}{c_t} &= \alpha \delta k_t^{\alpha - 1} E_t \left( \theta_{t+1} \frac{c_{t+1}^\eta}{c_{t+1}} \right) \\
c_t + k_t &= \theta_{t-1} k_{t-1}^\alpha \\
\theta_t &= \theta_{t-1} e^{\epsilon_t}
\end{align*}
\]

It is well known that when \(\eta = d = 1\), we have a closed form solution (Letttau, 2003):

\[
\begin{align*}
x_t(x_{t-1}, \epsilon_t) &\equiv D(x_{t-1}, \epsilon_t) \equiv \\
\left[ c_t(x_{t-1}, \epsilon_t) \right] &\equiv \\
\left[ k_t(x_{t-1}, \epsilon_t) \right] &\equiv \\
\left[ \theta_t(x_{t-1}, \epsilon_t) \right] &\equiv \\
\left[ (1 - \alpha \delta) \theta_t k_{t-1}^\alpha \right] \alpha \delta \theta_t k_{t-1}^\alpha \theta_{t-1}^\rho e^{\epsilon_t}
\end{align*}
\]

For mean zero iid \(\epsilon_t\) we can easily compute the conditional expectation of the model variables for any given \(\theta_{t+k}, k_{t+k}\):

\[
\begin{align*}
D(x_{t+k+1}) &\equiv \\
\left[ E_t(c_{t+k+1} | \theta_{t+k}, k_{t+k}) \right] &\equiv \\
\left[ E_t(k_{t+k+1} | \theta_{t+k}, k_{t+k}) \right] &\equiv \\
\left[ E_t(\theta_{t+k+1} | \theta_{t+k}, k_{t+k}) \right] &\equiv \\
\left[ (1 - \alpha \delta) k_{t+k}^\alpha e^{\frac{\rho}{2} \theta_{t+k}^2} \right] &\equiv \\
\left[ \alpha \delta k_{t+k}^\alpha e^{\frac{\rho}{2} \theta_{t+k}^2} \right] &\equiv \\
\left[ e^{\frac{\rho}{2} \theta_{t+k}^2} \right] &\equiv \\
\left[ \frac{(1 - \alpha \delta) k_{t+k}^\alpha e^{\frac{\rho}{2} \theta_{t+k}^2}}{\alpha \delta k_{t+k}^\alpha e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv \\
\left[ \frac{e^{\frac{\rho}{2} \theta_{t+k}^2}}{e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv \\
\left[ \frac{e^{\frac{\rho}{2} \theta_{t+k}^2}}{e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv \\
\left[ \frac{e^{\frac{\rho}{2} \theta_{t+k}^2}}{e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv \\
\left[ \frac{e^{\frac{\rho}{2} \theta_{t+k}^2}}{e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv \\
\left[ \frac{e^{\frac{\rho}{2} \theta_{t+k}^2}}{e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv \\
\left[ \frac{e^{\frac{\rho}{2} \theta_{t+k}^2}}{e^{\frac{\rho}{2} \theta_{t+k}^2}} \right] &\equiv
\end{align*}
\]

For any given values of \(k_{t-1}, \theta_{t-1}, \epsilon_t\), the model decision rule, \((D)\), and decision rule conditional expectation, \((\mathbb{D})\), can generate a conditional expectations path forward from any given initial \(x_{t-1}\) and \(\epsilon_t\):

\[
\begin{align*}
x_t(x_{t-1}, \epsilon_t) &= D(x_{t-1}, \epsilon_t) \\
x_{t+k+1}(x_{t-1}, \epsilon_t) &= \mathbb{D}(x_{t+k}(x_{t-1}, \epsilon_t))
\end{align*}
\]

and corresponding paths for \(\{z_{1t}(x_{t-1}, \epsilon_t), z_{2t}(x_{t-1}, \epsilon_t), z_{3t}(x_{t-1}, \epsilon_t), \ldots\}\)

\[
z_{t+k} \equiv H_{-1} x_{t+k-1} + H_0 x_{t+k} + H_1 x_{t+K+1}
\]

Formula (2) requires that

\[
x(x_{t-1}, \epsilon_t) = B \begin{bmatrix} c_{t-1} \\ k_{t-1} \\ \theta_{t-1} \end{bmatrix} + \phi \psi_t \epsilon_t + (I - F)^{-1} \phi \psi_c + \sum_{\nu=0}^{\infty} F^\nu \phi z_{t+\nu}(k_{t-1}, \theta_{t-1}, \epsilon_t)
\]

\(^6\)Here, I set their \(\beta = 1\) and do not discuss quasi-geometric discounting or time-inconsistency.
Figure 3: model variable values and z values

Figure 4: RBC Known Solution: Truncation Approximate Error Versus Actual: "Almost Arbitrary" Linear Reference Model

For example, using $d = 1$ and the following parameter values and using the arbitrary linear reference model (3) we can generate a series representation for the model solutions. With

$$\begin{bmatrix} k_{t-1} \\ \theta_{t-1} \\ \epsilon_t \end{bmatrix} = \begin{bmatrix} 0.18 \\ 1.1 \\ 0.01 \end{bmatrix}$$

(4)

The left panel of Figure 3 shows, from top to bottom, the paths of $\theta_t, c_t,$ and $k_t$ from the initial values given in Equation 4. The right panel shows the paths for the $z_t$ variables associated with the linear reference model. The orange line corresponds to $z_{1t}(x_{t-1}, \epsilon_t)$, the blue line corresponds to $z_{2t}(x_{t-1}, \epsilon_t)$ and the green line corresponds to $z_{3t}(x_{t-1}, \epsilon_t)$.

By including enough terms we can compute the solution for $x_t$ exactly. Figure 4 shows the impact that truncating the series has on approximation of the time $t$ values of the state variables. The approximated magnitude for the error, $B_n$, shown in red is again very pessimistic compared to the actual error, $Z_n = \|x_t - x_t^*\|_\infty$, shown in blue. With enough terms, even using an "almost" arbitrarily chosen linear model, the series approximation provides a machine precision accurate value for the time $t$ state vector. Thus we have wide latitude in choosing a linear reference model.

Figure 5 shows that using a linearization that better tracks the nonlinear model paths, improves the approximation $Z_n$, shown in blue and the approximate error $B_n$ shown in red. Using the linearization of the RBC model around the ergodic mean produces a tighter but
still pessimistic bound on the errors for the initial state vector. Again, the first few terms make most of the difference in approximating the value of the state variables.

### 3.2 A Model Specification Constraint

For convenience of notation in what follows, we will focus on models built up from components of the form

\[ h_i(x_{t-1}, x_t, x_{t+1}, \epsilon_t) = h_{i0}^{\text{det}}(x_{t-1}, x_t, \epsilon_t) + \sum_{j=1}^{p_i} h_{ij}^{\text{det}}(x_{t-1}, x_t, \epsilon_t) h_{ij}^{\text{nondet}}(x_{t+1}) = 0 \]

This is a very broad class of models including most widely used macroeconomics models. This constraint will make it easier to write down and manipulate the conditional expectations expressions in the description of the algorithms in subsequent sections.

For example, the Euler equations for the neoclassical growth model can be written as

\[
\begin{align*}
    h_{10}^{\text{det}}(\cdot) &= \frac{1}{c_t^\theta}, & h_{11}^{\text{det}}(\cdot) &= \alpha \delta k_t^{\alpha-1}, & h_{11}^{\text{nondet}}(\cdot) &= E_t \left( \frac{\theta_{t+1}}{c_{t+1}^\theta} \right) \\
    h_{20}^{\text{det}}(\cdot) &= c_t + k_t - \theta_t k_{t-1}^\theta, & h_{21}^{\text{det}}(\cdot) &= 0 \\
    h_{30}^{\text{det}}(\cdot) &= \ln \theta_t - (\rho \ln \theta_{t-1} + \epsilon_t), & h_{31}^{\text{det}}(\cdot) &= 0
\end{align*}
\]

Since we need to compute the conditional expectation of nonlinear expressions, this setup will make it possible for us to use auxiliary variables to correctly compute the required expected values. We will be working with models where expectations are computed at time \( t \), with \( \epsilon_t \) known. We can construct a linear reference model for the modified RBC system by augmenting the RBC model with the equation

\[ N_t = \frac{\theta_t}{c_t} \]

substituting \( N_{t+1} \) for \( \frac{\theta_{t+1}}{c_{t+1}} \) in the first equation and linearizing the RBC model about the ergodic mean.
This leads to:

\[
\begin{bmatrix}
H_{-1} & H_0 & H_1
\end{bmatrix} =
\begin{bmatrix}
0. & 0. & 0. & 0. & -7.6986 & 9.47963 & 0. & 0. & 0. & -0.999 & 0. \\
0. & -1.05263 & 0. & 0. & 1. & 1. & 0. & -0.547185 & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 7.7663 & 0. & 1. & -2.77463 & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & -0.949953 & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\
\end{bmatrix}
\]

with

\[
\psi_e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \psi_z = I
\]

These coefficients produce a unique stable linear solution.

\[
B = \begin{bmatrix}
0. & 0.692632 & 0. & 0.342028 \\
0. & 0.36 & 0. & 0.177771 \\
0. & -5.33763 & 0. & 0. \\
0. & 0. & 0. & 0.949953 \\
\end{bmatrix}, \phi = \begin{bmatrix}
-0.0444237 & 0.658 & 0. & 0.360048 \\
0.0444237 & 0.342 & 0. & 0.187137 \\
0.342342 & -5.07075 & 1. & 0. \\
0. & 0. & 0. & 0.1 \\
\end{bmatrix}
\]

\[
F = \begin{bmatrix}
0. & 0. & -0.0443793 & 0. \\
0. & 0. & 0.0443793 & 0. \\
0. & 0. & 0.342 & 0. \\
0. & 0. & 0. & 0. \\
\end{bmatrix}
\]

\[
\psi_c = \begin{bmatrix}
-3.7735 \\
-0.197184 \\
2.77741 \\
0.0500976 \\
\end{bmatrix}
\]

Recomputing the truncation errors with the expanded model produces nearly identical results for the variable approximation errors.

## 4 Approximating Model Solution Errors

This section shows how to write a series representation for any proposed model solution and to characterize the series representation for a typically unknown solution in such a way that one can use the model equation errors to construct an approximate error for the norm of the distance between the proposed and the exact solution.

This work contrasts with the analysis provided in [Judd et al. (2017); Peralta-Alva and Santos (2014); Santos and PeraltaâȂŔAlva (2005); Santos (2000)]. Their approaches identify Euler Equation Errors, but use statistical techniques to estimate the impact of these errors on solution accuracy. Here I provide a formula for the approximate error that does not require statistical inference.

Consider a bounded region, \( \mathcal{A} \), that contains all iterated conditional expectations paths. By bounding the largest deviation in the paths for the \( \Delta z_t^p \) we can bound the largest deviation in a proposed solution from the exact solution.

Given an exact solution \( x_t^* = g^*(x_{t-1}, \epsilon_t) \) define the conditional expectations function,

\[
G^*(x) \equiv \mathcal{E} [g^*(x, \epsilon)]
\]
then with
\[ E_t^*_{t+1} = G^*(g^*(x_{t-1}, \epsilon_t)) \]
we have
\[
\begin{align*}
M(x_{t-1}^*, x_t^*, E_t x_{t+1}^*, \epsilon_t) &= 0 \quad \forall (x_{t-1}, \epsilon_t) \\
x_t^*(x_{t-1}, \epsilon_t) &\in \mathbb{R}^L \quad \|x_t^*(x_{t-1}, \epsilon_t)\|_2 \leq \delta \quad \forall t > 0
\end{align*}
\]

Now consider a proposed solution for the model, \( x_t^p = g^p(x_{t-1}, \epsilon_t) \) define \( G^p(x) \equiv \mathcal{E}[g^p(x, \epsilon)] \) so that
\[
\begin{align*}
E_t x_{t+1}^p &= G^p(g^p(x_{t-1}, \epsilon_t)) \\
e_t^p(x_{t-1}, \epsilon) &\equiv M(x_{t-1}, x_t^p, E_t x_{t+1}^p, \epsilon_t)
\end{align*}
\]

\[
\|x_t^p(x_{t-1}, \epsilon) - x_t^p(x_{t-1}, \epsilon)\|_{\infty} \leq \max_{\{x_{t-1}, \epsilon\}} \left\| (I - F)^{-1} \phi M(x_{t-1}, g^p(x_{t-1}, \epsilon), G^p(g^p(x_{t-1}, \epsilon), \epsilon)) \right\|_2
\]
\[
\text{which can be approximated by}
\max_{x_{t-1}, \epsilon_t} (\phi e_t^p(x_{t-1}, \epsilon))^2
\]

For proof see Appendix A

5 Improving Proposed Solutions

5.1 Generalization and Practical Considerations for Applying the Series Formula

To fix notation, assume that our model is characterized by a countable number of equations systems. I will require that, for any given \((x_{t-1}, \epsilon_t)\), this collection of systems of equations produces a unique solution for \(x_t\). Using the convention described in section [3.2], we will include enough auxiliary variables so that we can correctly compute expected values for model variables by applying the law of iterated expectations.

It will be convenient for describing the algorithms to write the model in the form
\[
\{\{C_1, \ldots, C_M\}, d\}
\]

where, each
\[
C_i \equiv \{(b_i, M_i(x_{t-1}, \epsilon, x_t, \mathcal{E}[x_{t+1}]) = 0, c_i)\}
\]
represents a set of model equations, \( M_i \), with Boolean valued gate, \( b_i, c_i \). In the algorithm’s implementation, the \( b_i \) will be a boolean function precondition and the \( c_i \) will be a boolean function postcondition for a given equation system. The \( d_i \) will be a function for choosing
among the candidate \( x_t(x_{t-1}, \epsilon_t) \) values. Thus, we will have an equation system and corresponding gatekeeper logical expressions indicating which equation system is in force for producing the unique solution for a given \( x_{t-1}, \epsilon_t \). I will only consider time invariant model solutions \( x_t = x(x_{t-1}, \epsilon) \). Given a decision rule \( x(x_{t-1}, \epsilon) \), denote its conditional expectation \( X(x_{t-1}) \equiv \mathbb{E}[x(x_{t-1}, \epsilon)] \).

Examples of such systems include the simple RBC model from Section 3.2. Other examples include models with occasionally binding constraints with solutions exhibiting complementary slackness, or models with regime switching. See the appendix for an example of a specification for occasionally binding constraints (B.1) and for regime switching (B.2).

5.2 An Informal Characterization of the Algorithm

It may be worthwhile at this point to characterize how the approach I propose here differs from what is typically done for solving dynamic stochastic models. As with the standard approach I characterize the solutions using a linearly weighted sums of orthogonal polynomials and solve the model equations at predetermined collocation points. However, in this new algorithm, I augment the model Euler equations with equations based on \( 2 \) that link the conditional expectations for future values in the proposed solution with the time \( t \) value of the model variables. As a result, the algorithm determines linearly weighted polynomials characterizing the trajectories for the traditional variables \( x_t(x_{t-1}, \epsilon_t) \) as well as new \( z_t(x_{t-1}, \epsilon_t) \) variables. These new constraints help keep proposed solutions bounded during the algorithmic iterations thus improving algorithm reliability.

5.3 Algorithm Overview

I closely follow the techniques outlined in Judd et al. (2013, 2014). I use their an-isotropic Smolyak Lagrange interpolation with adaptive domain. I also precompute all of the conditional expectations for the integrals as outlined in \( ? \) so that the time \( t \) computations are all deterministic.

The algorithm uses a proposed decision rule characterized by a linearly weighted sum of Chebyshev polynomials with precomputed integrals. Since the orthogonal polynomials have been integrated, the linearly weighted sum provides the conditional expectations function needed for computing conditional expectations paths from each collocation point forward. This provides the expected future values for the \( z_t \) needed for the series representation. The algorithm then solves a deterministic problem at time \( t \) to improve the proposed solution. The algorithm can easily accommodate inequality constraints or regime switching.\(^7\)

\(^7\)See sections [B.1] and [B.2] characterizing models with regime switching.
4. decide upon the ranges for variables and the degrees of approximation for the an-
iso
tropic Smolyak polynomial representation
5. decide upon $K$, the number of conditional expectations function recursive iterations. Fewer means more truncation error. In effect, the algorithm imposes nonlinear con-
straints the for number of period specified and uses the linear reference model there-
after (ie $z_{t+N} = 0, \forall N > K$)

5.3.1 Single Equation System Case
For now, consider the single model equation system case. A description of the actual, multi-
system implementation follows.

$$
M(x_{t-1}, x_t, E[x_{t+1}], \epsilon) = 0.
$$

We seek

$$
M(x_{t-1}, x^*(x_{t-1}, \epsilon_t), X^*(x^*(x_{t-1}, \epsilon_t)), \epsilon) = 0 \ \forall (x_{t-1}, \epsilon_t).
$$

Given a proposed model solution

$$
x_t = x^p(x_{t-1}, \epsilon_t)
$$

compute

$$
X^p(x_{t-1}) \equiv E[x^p(x_{t-1}, \epsilon_t)].
$$

We will use a linear reference model $L \equiv \{H, \psi, \psi_c; B, \phi, F\}$ to construct a series of $z^p$ functions that help improve the accuracy of the proposed solution.

We can define functions $z^p, Z^p$ by

$$
z^p(x_{t-1}, \epsilon) \equiv H \begin{bmatrix}
x_{t-1} \\
x^p(x_{t-1}, \epsilon) \\
X^p(x(x_{t-1}, \epsilon))
\end{bmatrix} + \phi_c \epsilon_t + \phi_c
$$

$$
Z^p(x_{t-1}) \equiv E[z^p(x_{t-1}, \epsilon)].
$$

• Algorithm loop begins here. Define conditional expectations paths for $x_t, z_t$

$$
x_{t+k+1} = X(x_{t+k}), \ z_{t+k+1} = Z(x_{t+k}) \ \forall k \geq 0
$$

Using the $z^p(x_{t-1}, \epsilon)$ Series
• We get expressions for $x_t, E[x]_{t+1}$ consistent with $L$ and the conditional expectations path

$$
X_t = B x_{t-1} + \phi \psi_c \epsilon + (I - F)^{-1} \phi \psi_c + \sum_{\nu=0}^{\infty} F^\nu \phi Z(x_{t+\nu})
$$

$$
E[X_{t+1}] = B X_t + (I - F)^{-1} \phi \psi_c + \sum_{\nu=0}^{\infty} F^\nu \phi Z(x_{t+\nu}) \ \forall t, k \geq 0
$$

• Use the model equations, $M(x_{t-1}, x_t, E[x_{t+1}], \epsilon) = 0$ and $x_t(x_{t-1}, \epsilon_t) = X_t(x_{t-1}, \epsilon_t)$

to determine $x^p(x_{t-1}, \epsilon), z^p(x_{t-1}, \epsilon)$

• $x^p(x_{t-1}, \epsilon) = x^p(x_{t-1}, \epsilon), z^p(x_{t-1}, \epsilon) = z^p(x_{t-1}, \epsilon) – Repeat loop.$
5.3.2 Multiple Equation System Case

An outline of the current Mathematica implementation of the algorithm follows. Table 1 provides notation. Figure 6 provides a graphic depiction of the function call tree. For more algorithmic detail, see Appendix C. The current implementation has a couple of atypical features. Many of the calculations exploit Mathematica’s capability to blend numerical and symbolic computation and to easily use functions as arguments for other functions. In addition, the implementation exploits parallelism at several points. The parallel features apply whether or not one employs the the traditional technique or the approach suggested here. Below, I note where these features play a role.

Figure 6: Function Call Tree

nestInterp — Computes a sequence of decision rule and conditional expectation rule functions terminating when the evaluation of the decision rule functions at the tests points no longer changes much.

doInterp — Symbolically generates functions that return a unique $x_t$ for any value of $(x_{t-1}, \epsilon_t)$ for the family model equations, $\{\{C_1, \ldots, C_M\}, d\}$ and uses these functions to construct interpolating functions approximating the decision rules.

genFindRootFuncs — The function can use 2 or omit it thus implementing the usual approach for solving these models.

The routine symbolically generates a collection of function triples and an outer model solution selection function. Each of the function constructions can be done in parallel. Each of the triples returns the boolean gate values and a unique value for $x_t$
for any given value of \((x_{t-1}, \epsilon_t)\) the selection function chooses one solution from the set of model equations. The routine subsequently uses these functions to construct interpolating functions approximating the decision rules. Subcomponents of this step exploit parallelism.

**genLilXkZkFunc** — Symbolically creates a function that uses an initial \(Z_t\) path to generate a function of \((x_{t-1}, \epsilon_t, z_t)\) providing (potentially symbolic) inputs

\[
\begin{bmatrix}
  x_{t-1} \\
  x_t \\
  E_t(x_{t+1}) 
\end{bmatrix}
\]

for model system equations \(M\). This routine provides the information for constructing \(x_t(x_{t-1}, \epsilon_t)\) using the series expression \(2\).

**makeInterpFuncs** — Solves model equations at collocation points producing interpolation data and subsequently returns the interpolating functions for the decision rule and decision rule conditional expectation. This can be done in parallel.

### 5.4 Approximating the Known Solution: \(U(c) = \log(c)\)

This subsection uses the series representation to compute the solution for the case when the exact solution is known and to compare the various approximations to the known actual solution. In what follows, both the traditional and the new approach use an-isotropic Smolyak polynomial function approximation with precomputed expectations. Figure 7 characterizes the use of the parallelepiped transformation described in [Judd et al. (2013)](#). The left panel shows the 200 values of \(k_t\) and \(\theta_t\) resulting from a stochastic simulation of the the known decision rule. The right panel shows the transformed variables that constitute an improved set of variables for constructing function approximation values.

\[
\bar{X} = \begin{bmatrix} 0.192359 \\ 0.00585118 \\ \sigma_X = \begin{bmatrix} 1.01803 \\ 0.0206618 \\ 0.01 \end{bmatrix}
\end{bmatrix}
\]

\[
\begin{bmatrix} 0.00585118 \\ 0.0206618 \\ 0.01 \end{bmatrix}
\]
\[ V = \begin{pmatrix} -0.707107 & -0.707107 & 0. \\ -0.707107 & 0.707107 & 0. \\ 0. & 0. & 1. \end{pmatrix} \]

\[ \text{max } U = \begin{pmatrix} 2.80545 \\ 0.378337 \\ 3. \end{pmatrix} \]

\[ \text{min } U = \begin{pmatrix} -4.00456 \\ -0.36473 \\ -3. \end{pmatrix} \]

6 Conclusions

This paper introduces a new series representation for bounded time series and shows how to apply the series for representing bounded time invariant maps. This representation proves useful because the solutions for many dynamic stochastic economic models fall in this class. The series representation plays a strategic role in developing a formula for approximating the errors in dynamic model solution decision rules. The paper also shows how to augment traditional “Euler Equation” model solution methods with constraints reflecting how the updated conditional expectation paths impact the time t solution values. The current prototype was written in Mathematica and is available on GitHub.

There are many directions to pursue in future work. I have begun work developing a Julia version. In writing Mathematica code I applied a functional programming style while blending numeric and symbolic calculations. Along with Mathematica, Julia is also a homoiconic language where functions are first order objects. But, I suspect that the symbolic numeric balance may be different as I implement the code in Julia. Indeed, perhaps the Julia code will be completely numeric. In that implementation, I will investigate computational efficiency issues and further exploit the high degree of parallelism available in the algorithm.

References
A Error Approximation Formula Proof

We consider time invariant maps such as often arise from solving an optimization problem codified in a systems of equations. In what follows, we construct an error approximation for proposed model solutions. Consider a bounded region, \( A \), where all iterated conditional expectations paths remain bounded. Given an exact solution \( x^*_{t} = g^*(x_{t-1}, \epsilon_t) \) define the conditional expectations function,

\[
G^*(x) \equiv \mathcal{E}[g^*(x, \epsilon)]
\]

then with

\[
E_t x^*_{t+1} = G^*(g^*(x_{t-1}, \epsilon_t))
\]

we have

\[
\mathcal{M}(x^*_{t-1}, x^*_t, E_t x^*_{t+1}, \epsilon_t) = 0 \ \forall (x_{t-1}, \epsilon_t)
\]

In words, the exact solution exactly satisfies the model equations. Using \( G^* \) and \( L \), construct the family of trajectories and corresponding \( z^*_t(x_{t-1}, \epsilon) \)

\[
x^*_t(x_{t-1}, \epsilon_t) \in \mathbb{R}^L \quad ||x^*_t(x_{t-1}, \epsilon_t)||_2 \leq \bar{X} \ \forall t > 0
\]

\[
z^*_t(x_{t-1}, \epsilon_t) \equiv H_{-1} x^*_t(x_{t-1}, \epsilon_t) + H_0 x^*_t(x_{t-1}, \epsilon_t) + H_1 x^*_t(x_{t-1}, \epsilon_t).
\]

Consequently, the exact solution has a representation given by

\[
x^*_t(x_{t-1}, \epsilon) = B x_{t-1} + \phi \psi_t \epsilon + (I - F)^{-1} \phi \psi c + \sum_{\nu=0}^{\infty} F^* \phi z^*_t(x_{t-1}, \epsilon)
\]

and

\[
\mathcal{E}[x^*_{t+1}(x_{t-1}, \epsilon)] = B x^*_{t+k} + \sum_{\nu=0}^{\infty} F^\nu \phi \mathcal{E}[z^*_{t+1+\nu}(x_{t-1}, \epsilon)] + (I - F)^{-1} \phi \psi c
\]

with

\[
\mathcal{M}(x_{t-1}, x^*_t, E_t x^*_{t+1}, \epsilon_t) = 0 \ \forall (x_{t-1}, \epsilon_t)
\]

Now consider a proposed solution for the model, \( x^p_t = g^p(x_{t-1}, \epsilon_t) \) define \( G^p(x) \equiv \mathcal{E}[g^p(x, \epsilon)] \) so that

\[
E_t x^p_{t+1} = G^p(g^p(x_{t-1}, \epsilon_t))
\]

\[
\epsilon^p_t(x_{t-1}, \epsilon) \equiv \mathcal{M}(x_{t-1}, x^p_t, E_t x^p_{t+1}, \epsilon_t)
\]

19
By construction, the conditional expectations paths will all be bounded in the region $A^p$. Using $G^p$ and $L$ construct the family of trajectories and corresponding $z_t^p(x_{t-1}, \epsilon)$. 

$$x_t^p(x_{t-1}, \epsilon_t) \in \mathbb{R}^k \quad \|x_t^p(x_{t-1}, \epsilon_t)\|_2 \leq \bar{X} \quad \forall t > 0$$

$$z_t^p(x_{t-1}, \epsilon_t) \equiv H_{-1}x_{t-1}^p(x_{t-1}, \epsilon_t) + H_0x_t^p(x_{t-1}, \epsilon_t) + H_1x_{t+1}^p(x_{t-1}, \epsilon_t).$$

The proposed solution has a representation given by

$$x_t^p(x_{t-1}, \epsilon) = Bx_{t-1} + \phi\psi_\epsilon \epsilon + (I - F)^{-1}\phi\psi_c + \sum_{\nu=0}^{\infty} F^s\phi z_{t+\nu}^p(x_{t-1}, \epsilon)$$

and

$$\mathcal{E}[x_{t+1}^p(x_{t-1}, \epsilon)] = Bx_{t+k} + \sum_{\nu=0}^{\infty} F^{\nu}\phi z_{t+1+\nu}^p(x_{t-1}, \epsilon) + (I - F)^{-1}\phi\psi_c$$

with

$$e_t^p(x_{t-1}, \epsilon) \equiv M(x_{t-1}, x_t^p, E_t x_{t+1}^p, \epsilon_t)$$

$$x_t^*(x_{t-1}, \epsilon) - x_t^p(x_{t-1}, \epsilon) = \sum_{\nu=0}^{\infty} F^s\phi(z_{t+\nu}^*(x_{t-1}, \epsilon) - z_{t+\nu}^p(x_{t-1}, \epsilon))$$

$$\Delta z_{t+\nu}^p(x_{t-1}, \epsilon_t) \equiv (z_{t+\nu}^*(x_{t-1}, \epsilon) - z_{t+\nu}^p(x_{t-1}, \epsilon))$$

$$x_t^*(x_{t-1}, \epsilon) - x_t^p(x_{t-1}, \epsilon) = \sum_{\nu=0}^{\infty} F^s\phi \Delta z_{t+\nu}^p(x_{t-1}, \epsilon_t)$$

$$\|x_t^*(x_{t-1}, \epsilon) - x_t^p(x_{t-1}, \epsilon)\|_\infty \leq \sum_{\nu=0}^{\infty} F^s\phi \|\Delta z_{t+\nu}^p(x_{t-1}, \epsilon_t)\|_\infty$$

By bounding the largest deviation in the paths for the $\Delta z_t^p$ we can bound the largest difference in $x_t^p$. The exact solution satisfies the model equations exactly. The error associated with the proposed solution leads to a conservative bound on the largest change in $z$ needed to match the exact solution.

---

The algorithm presented below for finding solutions provides a mechanism for generating such proposed solutions.

Since the future values are probability weighted averages of the $\Delta z_t^p$ values for the given initial conditions and the condition expectations paths remain in the region $A^p$, the largest error for $\Delta z_{t+k}^p$ are pessimistic bounds for the errors from the conditional expectations path.
\[ \Delta z_t \leq \max_{\{x,-,\epsilon\}} \| \phi \mathcal{M}(x_-, g^p(x_-, \epsilon), G^p(g^p(x_-, \epsilon), \epsilon)) \|_2 \]

\[ \| x_t^*(x_{t-1}, \epsilon) - x_t^p(x_{t-1}, \epsilon) \| \leq \max_{\{x,-,\epsilon\}} \| (I - F)^{-1} \phi \mathcal{M}(x_-, g^p(x_-, \epsilon), G^p(g^p(x_-, \epsilon), \epsilon)) \|_2 \]

\[
\begin{align*}
z_t^* &= H_0 x_{t-1}^* + H_0 x_t + H_0 x_{t+1}^* \\
z_t^p &= H_0 x_{t-1}^p + H_0 x_t^p + H_0 x_{t+1}^p \\
0 &= \mathcal{M}(x_{t-1}^*, x_t^*, x_{t+1}^*, \epsilon_t) \\
e_t^p(x_{t-1}, \epsilon) &= \mathcal{M}(x_{t-1}^p, x_t^p, x_{t+1}^p, \epsilon_t) \\
\max_{x_{t-1}, \epsilon} (\| \phi \Delta z_t \|_2)^2 &= \max_{x_{t-1}, \epsilon} (\phi(H_0(x_t^p - x_t^*) + H_+(x_{t+1}^p - x_{t+1}^*)))^2
\end{align*}
\]

with

\[
\mathcal{M}(x_{t-1}^*, x_t^*, x_{t+1}^*, \epsilon_t) = 0
\]

\[
\Delta e_t^p(x_{t-1}, \epsilon) \approx \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} (x_t^* - x_t^p) + \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p)
\]

when \( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \) is non-singular, we can write

\[
(x_t^* - x_t^p) \approx \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right)^{-1} \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right)
\]

collecting results

\[
(\| \phi \Delta z_t \|_2)^2 =
\]

\[
(\phi(H_0 \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right) \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right) + H_+(x_{t+1}^p - x_{t+1}^*))^2 = \]

\[
(\phi(H_0 \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right) \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right) + H_+(x_{t+1}^p - x_{t+1}^*))^2 = \]

\[
(\phi(H_0 \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right) \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right) + H_+(x_{t+1}^p - x_{t+1}^*))^2 = \]

\[
(\phi(H_0 \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right) \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right) + H_+(x_{t+1}^p - x_{t+1}^*))^2 = \]

\[
(\phi(H_0 \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right) \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right) + H_+(x_{t+1}^p - x_{t+1}^*))^2 = \]

\[
(\phi(H_0 \left( \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \right) \left( \Delta e_t^p(x_{t-1}, \epsilon) - \frac{\partial \mathcal{M}(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} (x_{t+1}^* - x_{t+1}^p) \right) + H_+(x_{t+1}^p - x_{t+1}^*))^2 = \]

21
approximating the derivatives using the $H$ matrices

$$\frac{\partial M(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_t} \approx H_0$$

$$\frac{\partial M(x_{t-1}, x_t, x_{t+1}, \epsilon_t)}{\partial x_{t+1}} \approx H_1$$

produces

$$\max_{x_{t-1}, \epsilon_t} (\phi \Delta e_t^0(x_{t-1}, \epsilon))^2$$

### B Examples

#### B.1 Occasionally Binding Constraints

Stochastic dynamic non linear economic models increasingly embody occasionally binding constraints (OBC). Since Christiano and Fisher (2000) a host of authors have described a variety of approaches.\(^{10}\) Holden (2015); Guerrieri and Iacoviello (2015b); Benigno et al. (2009); Hintermaier and Koeniger (2010b); Brumm and Grill (2010); Nakov (2008); Haefke (1998); Nakata (2012); Gordon (2011); Billi (2011); Hintermaier and Koeniger (2010a); Guerrieri and Iacoviello (2015a)

Consider adding constraints the constraints:

$$I_t \geq \nu I_{ss}$$

We consider a model described in Maliar and Maliar (2005)\(^{11}\)

$$\max \left\{ u(c_t) + \mathbb{E}_t \sum_{t=0}^{\infty} \delta^{t+1} u(c_{t+1}) \right\}$$

$$c_t + k_t = (1 - d)k_{t-1} + \theta_t f(k_{t-1})$$

$$\theta_t = \theta_{t-1} e^{\epsilon_t}$$

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

$$u(c) = \frac{c^{1-\eta} - 1}{1-\eta}$$

$$L_t = \left\{ u(c_t) + \mathbb{E}_t \sum_{t=0}^{\infty} \delta^t u(c_{t+1}) \right\} +$$

$$\sum_{t=0}^{\infty} \left\{ \delta^t \lambda_t (c_t + k_t - ((1 - d)k_{t-1} + \theta_t f(k_{t-1}))) + \delta^t \mu_t (k_t - (1 - d)k_{t-1}) - \nu I_{ss} \right\}$$

\(^{10}\)The algorithms described Holden (2015) and Guerrieri and Iacoviello (2015b) also exploit the use of “anticipated shocks”, but do not use the comprehensive formula employed here.

\(^{11}\)Here, we set their $\beta = 1$ and do not discuss quasi-geometric discounting or time-inconsistency.
so that we can impose

\[ I_t = (k_t - (1 - d)k_{t-1}) \geq v I_{ss} \]

The first order conditions become

\[
\frac{1}{c_t} + \lambda_t
\]

\[ \lambda_t - \delta \lambda_{t+1} \theta_{t+1} \alpha k_t^{(a-1)} - \delta(1 - d) \lambda_{t+1} + \mu_t - \delta(1 - d) \mu_{t+1} \]

For example, the Euler equations for the neoclassical growth model can be written as

\[
\text{if } (\mu > 0 \land (k_t - (1 - d)k_{t-1} - v I_{ss}) = 0) \]

\[
\lambda_t = \frac{1}{c_t} \\
c_t + I_t - \theta_t k_t^{\alpha} \\
N_t - \lambda_t \theta_t \\
\theta_t - e^{(\rho \ln(\theta_{t-1}) + \epsilon)} \\
\lambda_t + \mu_t - (\alpha k_t^{(a-1)} \delta N_{t+1} + \lambda_{t+1} \delta(1 - d) + \mu_{t+1} + \delta(1 - d) \mu_{t+1}) \\
I_t - (K_t - (1 - d)k_{t-1}) \\
\mu_t(k_t - (1 - d)k_{t-1} - v I_{ss})
\]

\[
\text{if } (\mu = 0 \land (k_t - (1 - d)k_{t-1} - v I_{ss}) \geq 0) \]

\[
\lambda_t = \frac{1}{c_t} \\
c_t + I_t - \theta_t k_t^{\alpha} \\
N_t - \lambda_t \theta_t \\
\theta_t - e^{(\rho \ln(\theta_{t-1}) + \epsilon)} \\
\lambda_t + \mu_t - (\alpha k_t^{(a-1)} \delta N_{t+1} + \lambda_{t+1} \delta(1 - d) + \mu_{t+1} + \delta(1 - d) \mu_{t+1}) \\
I_t - (K_t - (1 - d)k_{t-1}) \\
\mu_t(k_t - (1 - d)k_{t-1} - v I_{ss})
\]

**B.2 Regime Switching**

To apply the series formula, one must construct conditional expectations paths from each initial \( x(x_{t-1}, \epsilon_t) \). Consider the case when the transition probabilities \( p_{ij} \) are constant and do not depend on \( x_{t-1}, \epsilon_t, x_t \).
\[ E_t[x_{t+1}] = \sum_{j=1}^{N} p_{ij} E_t(x_{t+1}(x_t)|s_{t+1} = j) \]

\[ E_t[x_{t+2}] = \sum_{j=1}^{N} p_{ij} E_t(x_{t+2}(x_{t+1})|s_{t+2} = j) \]

If we know the current state we can use the known conditional expectation function for the state:

\[
\begin{bmatrix}
E_t(x_{t+1}(x_t)|s_t = 1) \\
\vdots \\
E_t(x_{t+1}(x_t)|s_t = N)
\end{bmatrix}
\]

For the next state we can compute expectations if the errors are independent

\[
\begin{bmatrix}
E_t(x_{t+2}(x_t)|s_t = 1) \\
\vdots \\
E_t(x_{t+2}(x_t)|s_t = N)
\end{bmatrix} = (P \otimes I) \begin{bmatrix}
E_t(x_{t+2}(x_{t+1})|s_{t+1} = 1) \\
\vdots \\
E_t(x_{t+2}(x_{t+1})|s_{t+1} = N)
\end{bmatrix}
\]

Consider two states \( s_t \in 0,1 \) where the depreciation rates are different: \( d_0 > d_1 \)

\[
Prob(s_t = j|s_{t-1} = i) = p_{ij}
\]

\[
\begin{align*}
\text{if} (s_t = 0 \land \mu > 0 \land (k_t - (1 - d_0)k_{t-1} - vI_{ss}) = 0) \\
\lambda_t - \frac{1}{c_t} \\
c_t + k_t - \theta_t k_{t-1}^\alpha \\
N_t - \lambda_t \theta_t \\
\theta_t - e^{(\rho n(\theta_{t-1} + \epsilon))}
\end{align*}
\]

\[
\begin{align*}
\lambda_t + \mu_t - (\alpha k_t^{\alpha-1}) \delta N_{t+1} + \lambda_{t+1} \delta (1 - d_0) + \mu_{t+1} + \delta (1 - d_0) \\
I_t - (K_t - (1 - d_0) k_{t-1}) \\
\mu_t (k_t - (1 - d_0) k_{t-1} - vI_{ss})
\end{align*}
\]
if \( s_t = 0 \land \mu = 0 \land (k_t - (1 - d_0)k_{t-1} - vI_{ss}) \geq 0 \)

\[
\begin{align*}
\lambda_t &= \frac{1}{c_t} \\
\theta_t &= \frac{k_t - \theta_t k_{t-1}^\alpha}{N_t - \lambda_t \theta_t} \\
\theta_t &= \exp(\rho \ln(\theta_{t-1}) + \epsilon) \\
\end{align*}
\]

\[
\lambda_t + \mu_t - (\alpha k_t^{(\alpha-1)} \delta N_{t+1} + \lambda_{t+1} \delta(1 - d_0) + \mu_{t+1} + \delta(1 - d_0)) \\
I_t - (K_t - (1 - d_0)k_{t-1}) \\
\mu_t(k_t - (1 - d_0)k_{t-1} - vI_{ss})
\]

if \( s_t = 1 \land \mu > 0 \land (k_t - (1 - d_1)k_{t-1} - vI_{ss}) = 0 \)

\[
\begin{align*}
\lambda_t &= \frac{1}{c_t} \\
\theta_t &= \frac{k_t - \theta_t k_{t-1}^\alpha}{N_t - \lambda_t \theta_t} \\
\theta_t &= \exp(\rho \ln(\theta_{t-1}) + \epsilon) \\
\end{align*}
\]

\[
\lambda_t + \mu_t - (\alpha k_t^{(\alpha-1)} \delta N_{t+1} + \lambda_{t+1} \delta(1 - d_1) + \mu_{t+1} + \delta(1 - d_1)) \\
I_t - (K_t - (1 - d_1)k_{t-1}) \\
\mu_t(k_t - (1 - d_1)k_{t-1} - vI_{ss})
\]

if \( s_t = 1 \land \mu = 0 \land (k_t - (1 - d_1)k_{t-1} - vI_{ss}) \geq 0 \)

\[
\begin{align*}
\lambda_t &= \frac{1}{c_t} \\
\theta_t &= \frac{k_t - \theta_t k_{t-1}^\alpha}{N_t - \lambda_t \theta_t} \\
\theta_t &= \exp(\rho \ln(\theta_{t-1}) + \epsilon) \\
\end{align*}
\]

\[
\lambda_t + \mu_t - (\alpha k_t^{(\alpha-1)} \delta N_{t+1} + \lambda_{t+1} \delta(1 - d_1) + \mu_{t+1} + \delta(1 - d_1)) \\
I_t - (K_t - (1 - d_1)k_{t-1}) \\
\mu_t(k_t - (1 - d_1)k_{t-1} - vI_{ss})
\]
C  Algorithm Pseudo-code

\[ \mathcal{L} \equiv \{ H, \psi, \psi_c; B, \phi, F \} \]  The linear reference model

\[ x^p(x_{t-1}, \epsilon_t) \]  The proposed decision rule \( x_t \) components

\[ z^p(x_{t-1}, \epsilon_t) \]  The proposed decision rule \( z_t \) components

\[ X^p(x_{t-1}) \]  The proposed decision rule conditional expectations \( x_t \) components

\[ Z^p(x_{t-1}) \]  The proposed decision rule their conditional expectations \( z_t \) components

\( \kappa \)  One less than the number of terms in series representation \( (\kappa \geq 0) \)

\( S \)  Smolyak an-isotropic grid polynomials \( (p_1(x, \epsilon), \ldots, p_N(x, \epsilon)) \) and their conditional expectations \( (P_1(x), \ldots, P_N(x)) \)

\( T \)  Model evaluation points Neidereiter sequence in the model variable ergodic region.

\( \gamma \{ x(x_{t-1}, \epsilon_t), z(x_{t-1}, \epsilon_t) \} \) collects the decision rule components

\( \Gamma \{ x(x_{t-1}, \epsilon_t), z(x_{t-1}, \epsilon_t) \} \) collects the decision rule conditional expectations components

\( \mathcal{H} \{ \gamma, \Gamma \} \) collects decision rule information

\[ \{ \{ C_1, \ldots, C_M \}, d \} \]  The equation system \( \mathcal{R}^{3N_e+N_i+N_x} \rightarrow \mathcal{R}^{N_x} \)

\begin{verbatim}
input : (L, H^0, \kappa, \{\{ C_1, \ldots, C_M \}, d \}, S, T)
1  /* Compute a sequence of decision rule and conditional expectation rule
    function terminating when the evaluation of the functions at the
    tests points no longer changes much. Norm of the difference
    controlled by default (''normConvTol''->10^{-10}) */
2  NestWhile
    (Function(v, doInterp(L, v, \kappa, \{\{ C_1, \ldots, C_M \}, d \}, S)), H^0, notConvergedAt(v, T))
output: \{H^0, H^1, \ldots, H^c\}
\end{verbatim}

Algorithm 1: nestInterp

D  Other Computing?

- Table 1 out of place
- fix genInterpData checks all signatures
- Call tree, hyperlink?
- function names bold greek and function name reconcile.
input : $(\mathcal{L}, \mathbb{H}, \kappa, \{\{C_1, \ldots, C_M\}, d\}, S)$

1 /* Symbolically generates a collection of function triples and an outer
    model solution selection function. Each of triples returns the
    boolean gate values and a unique value for $x_t$ for any given value of
    $(x_{t-1}, \epsilon_t)$ one of the model equations and subsequently uses these
    functions to construct interpolating functions approximating the
decision rules. */

2 $\text{theFindRootFuncs} = \text{genFindRootFuncs}(\mathcal{L}, \mathbb{H}, \kappa, \{\{C_1, \ldots, C_M\}, d\})$

3 $\mathbb{H}^{i+1} = \text{makeInterpFuncs}(\text{theFindRootFuncs}, S)$

Algorithm 2: doInterp

---

input : $(\mathcal{L}, \mathbb{H}, \kappa, \{\{C_1, \ldots, C_M\}, d\}, S)$

1 /* Applies genFindRootWorker to each model component. Symbolically
    generates a collection of function triples and an outer model
    solution selection function. Each of the triples returns the
    boolean gate values and a unique value for $x_t$ for any given value of
    $(x_{t-1}, \epsilon_t)$ one from the set of model equations. It subsequently
    uses these functions to construct interpolating functions
    approximating the decision rules. Each of the function
    constructions can be done in parallel. */

2 $\text{theFuncOptionsTriples} =$
   Map(Function($v$, $\{v[1], \text{genFindRootWorker}(\mathcal{L}, \mathbb{H}, \kappa, v[2]), v[3]\}, \{C_1, \ldots, C_M\}$), $\mathbb{C}$)

output: $\{\text{theFuncOptionsTriples}, d\}$

Algorithm 3: genFindRootFuncs

---

input : $(\mathcal{L}, \mathbb{G}, \kappa, M)$

1 /* Symbolically constructs functions that return $x_t$ for a given $(x_{t-1}, \epsilon_t)$
   */

2 $\mathcal{Z} = \{Z_{t+1}, \ldots, Z_{t+k-1}\} = \text{genZsForFindRoot}(\mathcal{L}, x^p_t, \mathbb{G})$

3 $\text{modEqnArgsFunc} = \text{genLilXkZkFunc}(\mathcal{L}, \mathcal{Z})$

4 $\mathcal{X} = \{x_{t-1}, x_t, E_t(x_{t+1}), \epsilon_t\} = \text{modEqnArgsFunc}(x_{t-1}, \epsilon_t, z_t)$

5 $\text{funcOfXtZt} = \text{FindRoot}((x^p_t, z_t), \{M(\mathcal{X}), x_t - x^p_t\})$

6 $\text{funcOfXtm1Eps} = \text{Function}((x_{t-1}, \epsilon_t), \text{funcOfXtZt})$

output: $\text{funcOfXtm1Eps}$

Algorithm 4: genFindRootWorker
input : \((x_t^0, L, G, \kappa, M)\)

1 /* Symbolically compute the \(\kappa - 1\) future conditional \(Z\)'s vectors needed for the series formula. (empty list for \(\kappa = 1\) */  

2 \(X_t = x_t^0\) for \(i = 1, i < \kappa - 1, i + + \) do  

3 \(\{X_{t+i}, Z_{t+i}\} = G(Z_{t+i-1})\)

4 end

5 = \{(X_{t+1}, Z_{t+1}), \ldots, (X_{t+k-1}, Z_{t+k-1})\} = \text{genZsForFindRoot}(L, x_t^0, G)

Algorithm 5: genZsForFindRoot

input : \((L = \{H, \psi_c, \psi_c; B, \phi, F\})\)

1 /* Create functions that use the solution from the linear reference model for an initial proposed decision rule function and decision rule conditional expectation function. */  

2 \(\gamma(x, \epsilon) = \begin{bmatrix} Bx + (I - F)^{-1}\phi\psi_c + \psi_c\epsilon_t \\ 0_{N_x} \end{bmatrix}\)

3 \(G(x, \epsilon) = \begin{bmatrix} Bx + (I - F)^{-1}\phi\psi_c + \psi_c \\ 0_{N_x} \end{bmatrix}\)

4 \(\mathbb{H} = \{\gamma, G\}\)

output: \{\mathbb{H}\}

Algorithm 6: genBothX0Z0Funcs

input : \((L, \{Z_{t+1}, \ldots, Z_{t+k-1}\})\)

1 /* Symbolically creates a function that uses an initial \(Z_t\) path to generate a function of \((x_{t-1}, \epsilon_t, z_t)\) providing (potentially symbolic) inputs \((x_{t-1}, x_t, E_t(x_{t+1}), \epsilon_t)\), for model system equations \(M\) */  

2 \(fCon = fSumC(\phi, F, \psi_z, \{Z_{t+1}, \ldots, Z_{t+k-1}\})\)

xtVals = \text{genXtOfXtm1}(L, x_{t-1}, \epsilon_t, z_t, fCon)

xtpVals = \text{genXtp1OfXt}(L, xtVals, fCon)

fullVec = \{xtm1Vals, xtVals, xtp1Vals, epsVars\}

output: fullVec

Algorithm 7: genLilXkZkFunc

input : \((L, x_{t-1}, \epsilon_t, z_t, fCon)\)

1 /* Symbolically creates a function that uses an initial \(Z_t\) path to generate a function of \((x_{t-1}, \epsilon_t, z_t)\) providing (potentially symbolically) the \(x_t\) component of inputs \((x_{t-1}, x_t, E_t(x_{t+1}), \epsilon_t)\), for model system equations \(M\) */  

2 \(x_t = Bx_{t-1} + (I - F)^{-1}\phi\psi_c + \phi.\psi_z.\epsilon_t + \phi.\psi_z.z_t + FfCon\)

output: \(x_t\)

Algorithm 8: genXtOfXtm1
input : \( (L, x_{t-1}, f \text{Con}) \)

1 /* Symbolically creates a function that uses an initial \( Z_t \) path to
generate a function of \( (x_{t-1}, \epsilon_t, z_t) \) providing (potentially symbolically )
the \( x_{t+1} \) component of inputs \( (x_{t-1}, x_t, E_t(x_{t+1}), \epsilon_t) \), for model system
equations \( M \) */

2 \[ x_t = Bx_{t-1} + (I - F)^{-1} \phi \psi_c + \phi \psi \epsilon_t + \phi \psi z_t + f \text{Con} \]

output: \( x_t \)

Algorithm 9: \textit{genXtp10fXt}

input : \( (L, \{Z_{t+1}, \ldots, Z_{t+k-1}\}) \)

1 /* Numerically sums the \( F \) contribution for the series. */

2 \[ S = \sum F^{\nu-1} \phi Z_{t+\nu} \]

output: \( S \)

Algorithm 10: \textit{fSumC}

input : \( (\{\{C_1, \ldots, C_M\}, d\}, S) \)

1 /* Solves model equations at collocation points producing interpolation
data and subsequently returns the interpolating functions for the
decision rule and decision rule conditional expectation. This
routine also optionally replaces the approximations generated for
all ‘‘backward looking’’ equations with user provided pre-computed
functions. */

2 \[ \text{interpData} = \text{genInterpData}(\{\{C_1, \ldots, C_M\}, d\}, S) \]

\{\gamma, G\} = \text{interpDataToFunc(interpData, S)}

output: \{\gamma, G\}

Algorithm 11: \textit{makeInterpFuncs}

input : \( (\{\{C_1, \ldots, C_M\}, d\}, S) \)

1 /* Solves model equations at collocation points producing interpolation
data and subsequently returns the interpolating functions for the
decision rule and decision rule conditional expectation. This
routine also optionally replaces the approximations generated for
all ‘‘backward looking’’ equations with user provided pre-computed
functions. */

output: \{\gamma, G\}

Algorithm 12: \textit{genInterpData}

input : \( (\mathbb{C}, (x_{t-1}, \epsilon_t)) \)

1 /* Evaluate the model equations obeying pre and post conditions */

output: \( x_t \) or failed

Algorithm 13: \textit{evaluateTriple}
/* Computes a vector of Smolyak interpolating polynomials corresponding to the matrix of function evaluations. Each row corresponds to a vector of values at a collocation point. */

\textbf{Algorithm 14: interpDataToFunc}

- More information about other error formulae.
- multiple formulae?
- legend for figs throughout
- axes labels throughout
- eliminate colors in graphs throughout
- correct caption for ergodic values transformed
- mention genboth
- show solution accuracy for unknown solution
- show solution accuracy simple and OBC
- revise assess error calc or eliminate
- revise assess robust or eliminate
- just the first term.
- Proof for robustness and accuracy and parallelism gains.
- doPrepPoly
- calculate using their error?
- parallelism measure
- Large model multiple errors
- units for $z$
\[ \mathcal{L} \equiv \{ H, \psi_c, \psi_c^*; B, \phi, F \} \] The linear reference model

\[ x^p(x_{t-1}, \epsilon_t) \] The proposed decision rule \( x_t \) components

\[ z^p(x_{t-1}, \epsilon_t) \] The proposed decision rule \( z_t \) components

\[ X^p(x_{t-1}) \] The proposed decision rule conditional expectations \( x_t \) components

\[ Z^p(x_{t-1}) \] The proposed decision rule their conditional expectations \( z_t \) components

\( \kappa \) One less than the number of terms in series representation \( (\kappa \geq 0) \)

\( \mathbb{S} \) Smolyak an-isotropic grid polynomials \( (p_1(x, \epsilon), \ldots, p_N(x, \epsilon)) \) and their conditional expectations \( (P_1(x), \ldots, P_N(x)) \)

\( \mathbb{T} \) Model evaluation points Neidereiter sequence in the model variable ergodic region.

\( \gamma \) \( \{ x(x_{t-1}, \epsilon_t), z(x_{t-1}, \epsilon_t) \} \) collects the decision rule components

\( \mathbb{G} \) \( \{ x(x_{t-1}, \epsilon_t), z(x_{t-1}, \epsilon_t) \} \) collects the decision rule conditional expectations components

\( \mathbb{H} \) \( \{ \gamma, \mathbb{G} \} \) collects decision rule information

\( \{ \mathbb{C}_1, \ldots, \mathbb{C}_M \}, d \) The equation system \( \mathcal{R}^{3N_x + N_c + N_x} \to \mathcal{R}^{N_x} \)

Table 1: Algorithm Notation