Discussion Paper Series

Real-Time, Adaptive Learning via Parameterized Expectations

By

Michele Berardi and John Duffy

Centre for Growth and Business Cycle Research, Economic Studies,
University of Manchester, Manchester, M13 9PL, UK

Real-Time, Adaptive Learning via Parameterized Expectations[*]

Michele Berardi
University of Manchester
Manchester, M13 9PL
United Kingdom

John Duffy
University of Pittsburgh
Pittsburgh, PA 15260
United States

July 2010

Abstract

We explore real time, adaptive nonlinear learning dynamics in stochastic macroeconomic systems. Rather than linearizing nonlinear Euler equations where expectations play a role around a steady state, we instead approximate the nonlinear expected values using the method of parameterized expectations. Further we suppose that these approximated expectations are updated in real time as new data become available. We explore whether this method of real-time parameterized expectations learning provides a plausible alternative to real-time adaptive learning dynamics under linearized versions of the same nonlinear system.

Keywords: Rational expectations, learning, parameterized expectations, numerical methods.
JEL **Codes:** C62, D83.

1 Introduction

The macroeconomics learning literature (see, e.g. Evans and Honkapohja (2001)) has sought to provide a microfounded justification for the use of the rational expectations hypothesis. Specifically, this learning literature posits that agents do not initially possess rational expectations but are instead endowed with a perceived law of motion (PLM) characterizing the evolution of the endogenous variables in the system in which they operate. In many applications this PLM is correctly specified in the sense that it nests the rational expectations solution of the system as a special case. Thus, learning agents must simply determine the correct (rational expectations) parameterization of their PLM. In many learning models, agents are viewed as approaching this task as though they were econometricians: they use their PLM to form forecasts of future endogenous variables. Those forecasts interact with the equations of the system to generate actual realizations of the endogenous variables the agents were forecasting. The coefficients of the PLM are then updated in real time taking into account the new observations on the endogenous variables and the past forecast errors. The result is a mapping from perceptions to outcomes. In constructing this mapping, the common practice is to first linearize the nonlinear macroeconomic system around the rational expectations solution (of course, this is not an issue for linear rational expectations models). The reduced form linear specification of the macroeconomic model is the one that learning agents adopt as their PLM, which they use to form expectations of the future endogenous variables of the model.

Operationally, the approach using linear PLMs is straightforward; the resulting mapping from perceived to actual realizations will necessarily be linear and so stability can be assessed using standard techniques. Conceptually, however, this approach is problematic as it presumes that agents know the correct, linearized reduced form specification of the model. While such knowledge of the PLM could be guessed, in practice, the researcher obtains the linearized reduced form model by first solving the model under the assumption of rational expectations. Thus, following the same logic, the agents residing in the model would themselves have to solve the model under rational expectations *before* they could begin using the model to learn how to form rational expectations. We seek a more reasonable approach.

Specifically, we suppose that learning agents adopt the method of parameterized expectations which Marcet (1988) proposed as a means of finding solutions to stochastic, nonlinear rational expectations models. Under this approach, the conditional expectations found in the (nonlinear) Euler equations are approximated using flexible functional forms, typically polynomials. The parameters of those functional forms are then adjusted over time until the Euler equations using parameterized expectations provide a close fit to the historical time series data. Thus, this method completely avoids the practice of first linearizing the Euler equations of the model and then solving the linearized system under rational expectations.

The innovation of this paper is to suppose that the method of parameterized expectations developed to

find rational expectations solutions, is instead applied in *real time* by the agents who are seeking to learn a rational expectations solution to the model in which they operate. Specifically, the current parameterization of the conditional expectations of the Euler equation system is used to determine the current period choice (endogenous) variables. These variables are then added to the history of data against which the parameterized expectation function is judged; if a tolerance criterion is not met, further adjustments are made to the parameters of the approximating functions. Thus, agents in our model are simultaneously learning both the specification of the model to use for forecasting as well as the parameters of that forecast model in real-time. In this sense they act even more like econometricians in the sense that they are uncertain of their regression specification. Our aim is to assess the extent to which this method of real-time adaptive, nonlinear, parameterized expectations learning reliably leads agents to converge on the rational expectations solution of macroeconomic models and whether our approach is computationally more time intensive than the more standard approach using linearized macroeconomic models.

Our approach to learning lies within a framework that Evans et al. (2003) and Evans and Honkapohja (2006) have termed "Euler-equation" learning, as the agents in our model take the Euler equation as the fundamental primitive and, consequently, they make only *one*-step ahead forecasts at each date in time. This approach can be contrasted with the "infinite horizon learning" approach of Preston (2005) where, in each period, agents revise the sequence of forecasts they use over their remaining planning horizon. Both of these approaches examine stability under learning using learning agents' forecasts in *linearized* versions of the macroeconomic system. As we have stressed above, we are considering the question of stability under learning and model specification simultaneously and without resorting to linearization.[1]

2   The Method of Parameterized Expectation

The method of parameterized expectations was proposed by Marcet (1988) as a way to solve for the RE equilibrium in nonlinear models, by approximating an unknown nonlinear expectational function with a parametric function.[2] The general form of the model to be solve can be written as:

$$f\left(E_t[g(z_{t+1}, z_t)], z_t, z_{t-1}, \epsilon_t\right) = 0, \tag{1}$$

where f and g are known functions, $z_t \in \mathbb{R}^n$ is a vector of endogenous and exogenous variables describing the economy, and $\epsilon_t \in \mathbb{R}^s$ is a vector of innovations.

---

[1] We note that Evans and McGough (2009) have also questioned the standard practice of assuming boundedly rational learning agents *after* the model has been solved and linearized around the rational expectations steady state. Their "shadow-price learning" approach differs from our approach in that they impose bounded rationality *prior* to solving for the equilibrium of the model; specifically, agents in their model forecast one-step ahead values for shadow prices, which they then use to make current period decisions.

[2] For a detailed description of this method and its application to various macroeconomic models, see, e.g., Den Hann and Marcet (1990), Marcet and Marshall (1994) and Marcet and Lorenzoni (1999).

The general strategy is to approximate the expectational function $E_t[g(z_{t+1}, z_t)]$ with a flexible parametric function $\gamma(\theta, x_t)$,[3] where θ is the vector of parameters and x_t is the vector of state variables (usually including lagged endogenous and exogenous components) that summarize all available information, i.e.

$$E_t[g(z_{t+1}, z_t)] = E[g(z_{t+1}, z_t) \mid x_t].$$

Given the approximating function $\gamma(\theta, x_t)$ and an initial guess for θ, the idea is to find the vector θ that minimizes

$$||E_t[g(z_{t+1}, z_t)] - \gamma(\theta, x_t)||^2.$$

A common approach is to start with some initial guess θ_i (and initial values x_0), and use $\gamma(\theta_i, x_t)$ to generate a long series of data, $\{z_t\}_{t=1}^T$. Then use this data to run the nonlinear least squares regression

$$g(z_{t+1}, z_t) = \gamma(\theta_i, x_t) + u_t.$$

and obtain an estimate $\hat{\theta}$. Finally, update the guess for θ using this estimate:

$$\theta_{i+1} = \phi\hat{\theta} + (1 - \phi)\theta_i$$

where $\phi \in (0, 1)$ is a smoothing parameter. This process is repeatedly applied until a tolerance threshold τ is reached: $|\theta_{i+1} - \theta_i| < \tau$

While the method of PE was originally thought of as an off–line algorithm to be used to find a RE solution for the model, we think it is possible to interpret it as an on–line, or "real–time" algorithm used by agents in the model to learn the structure of the economy. We therefore propose to apply the PE algorithm as a recursive learning algorithm for agents when the equations they need to learn are nonlinear (and of an unknown form). We view agents' lack of precise knowledge regarding the functional form for conditional expectations as a core feature of their learning problem in nonlinear macroeconomic systems.

3 Learning in nonlinear models: an introduction

Learning in nonlinear models has received relatively little attention in the literature. Notable exceptions are Evans & Honkapohja (1995) and (2001, chapters 11 and 12), and Eusepi (2007). All these works, though, focus on steady states or k-cycles of the (stationary) nonlinear models and thus do not need to find an approximate solution for the expectational function. Specifically, given the nonlinear model

$$z_t = H(E_t G(z_{t+1,}\epsilon_{t+1}), \epsilon_t) \tag{2}$$

---

[3]Note that solutions are restricted to satisfy a recursive framework, i.e., the approximating function used in place of the conditional expectations is taken to be time-invariant.

these works find a solution of the form

$$z_t = H(\theta_j, \epsilon_t)$$

where $j = i + 1$, $i = 0, 1, 2, .., k - 1$, for $t \bmod k = i$ in case of a k-cycle ($k = 1$ in case of a steady state). As agents are assumed to know they are in a k-cycle, they learn separately k different values θ_j:[4] it is therefore not necessary to find an approximation to the expectational function $E_t G(.)$. Note that focusing only on steady states (or cycles) makes sense only when the structural model is i.i.d. (no lagged components in the structural equations, and the shoks are iid random variables), so that no transition dynamics can be expected to emerge.

Our aim in this paper is to analyze nonlinear learning in models where transition dynamics play an important role, as it is the case for models of the form (1): agents cannot assume to be in a steady state (or k-cycle), and they need to find an approximate solution for the expectational function $E_t G(.)$ based only on information currently available to them (current state variables and exogenous shocks)

$$E_t G(z_{t+1}, \epsilon_{t+1}) \simeq F(z_{t-1}, \epsilon_t; \theta),$$

where θ is the vector of parameters used to parameterized the function F.[5]

3.1 A toy model

We present now the general idea of real–time learning in nonlinear model using a toy model. A simplifying assumption we make here is that the (unique) state variable is purely exogenous, thus allowing us to compute the limit of distributions and to use stochastic approximation techniques to work out analytical solutions. This is not in general possible when the state space includes lagged endogenous variables, and we have to resort to numerical simulations.

The simple model we are going to use now is

$$y_t = E_t y_{t+1}^\alpha z_t^\beta \qquad (3)$$

$$z_t = \bar{z} + \rho z_{t-1} + e_t \qquad (4)$$

where e_t is i.i.d. with usual properties. For simplicity, we assume that when $E_t(.)$ are formed, agents can already observe z_t.

3.2 Solving with PE

The procedure to find a RE solution with the PE approach is the following:

---

[4]Given the structural form (2), θ_j are independently distributed over time and identically distributed for the same values of j when ϵ_t is iid.

[5]Chen and White (1998) analyse instead the case where agents learn $E_t G(.)$ nonparametrically.

1. Generate series for x_t and select initial values for $\theta's$.

2. Approximate $E_t y_{t+1}^\alpha$ with exponential polynomial

$$e^{\theta_0 + \theta_1 z_t + \theta_2 z_t^2} = e^{\theta' x_t}$$

where $\theta = [\theta_0 \ \theta_1 \ \theta_2]'$, $x_t = [1 \ z_t \ z_t^2]'$.

3. Using

$$y_t \simeq e^{\theta' x_t} z_t^\beta$$

generate $\{y_t\}_1^T$.

4. Imposing RE, compute $\{y_t^{RE}\}_1^T$ where

$$y_t^{RE} = y_{t+1}^\alpha z_t^\beta.$$

5. We need to find

$$\theta = \arg\min \frac{1}{T} \sum_{t=1}^T \| \left(y_{t+1}^{RE}\right)^\alpha - e^{\theta' x_t} \|^2$$

and therefore regress $log\left\{\left(y_{t+1}^{RE}\right)^\alpha\right\}_1^T$ on $x_t = [1 \quad z_t \quad z_t^2]'$ and find new values for $\theta's$.

6. Repeat 3-5 until convergence of estimates.

Simulations show that estimates for θ converge.[6] We can interpret the PEA as having agents learn across economies.

3.3 Solving with nonlinear RLS via PE

We now modify this procedure in order to implement real time learning in this model. Agents, instead of learning across economies, will now learn over time. By applying techniques from the recursive least squares learning literature, we recursively estimate the parameterized expectations model in order to implement real time learning. The procedure we follow is the following:

1. Set initial values for $\theta's$ and generate series for z_t.

2. Compute expectations

$$E_t y_{t+1}^\alpha \simeq e^{\theta_0 + \theta_1 z_t + \theta_2 z_t^2}$$

3. Insert expectations into structural model and compute actual law of motion (ALM):

$$y_t = e^{\theta_0 + \theta_1 z_t + \theta_2 z_t^2} z_t^\beta = e^{\theta' x_t} z_t^\beta$$

which implies the nonlinear ALM:

$$y_t^\alpha = e^{\alpha \theta' x_t} z_t^{\alpha\beta}.$$

---

[6]In the special case with $\alpha = 0$ and $\beta = 1$ (linear case), where an analitical solution is readily available, θ converge to the right values: (0 0 0).

4. Using the fact that, from step 2

$$E_{t-1}y_t^\alpha = e^{\theta' x_{t-1}},$$

update $\theta's$ using nonlinear RLS procedure (this is in fact a log linear specification; for other forms of nonlinearities, a nonlinear least squares recursive procedure should be implemented):

$$\theta_t = \theta_{t-1} + t^{-1}R_{t-1}^{-1}x_{t-1}(\log(y_t^\alpha) - \log(E_{t-1}y_t^\alpha)) \quad (5)$$

$$R_t = R_{t-1} + t^{-1}(x_t x_t' - R_{t-1}) \quad (6)$$

5. Repeat 2-4 until convergence.

We can analyze analytically the behavior of this stochastic recursive algorithm. Using the fact that

$$\log(y_t^\alpha) = \alpha\theta' x_t + \alpha\beta\log(z_t)$$

$$\log(E_{t-1}y_t^\alpha) = \theta' x_{t-1}$$

we can rewrite the error term in the stochastic recursive algorithm as

$$\alpha\beta\log(z_t) + \alpha\theta' x_t - \theta' x_{t-1}.$$

Using stochastic approximation techniques (see Evans and Honkapohja, 2001), we can compute the differential equation

$$\frac{d\theta}{d\tau} = (\alpha\rho - 1)\,\theta + \alpha\beta R^{-1}x\log(z)$$

that represents the limiting behaviour of (5) and whose stability is governed by the term $(\alpha\rho - 1)$. Therefore, for $\alpha < 1/\rho$ the REE is learnable (E-stable).[7]

Though here the equations are log-linear, the general idea is to apply recursive nonlinear least squares (RNLS) to a parameterized model meant to approximate the expectational equation. Nonlinear least squares can rarely be solved in closed form and the NLS estimates must be computed using numerical methods. This requires to optimize a nonlinear function using an iterative algorithm that repeatedly calculates new parameter values until the criterion function is minimized. This is usually done with a given set of data, until convergence of estimates (possibly to a local minimum).

4 An application to the optimal growth model

In this section we apply our approach to model real time learning by means of parameterized expectations to the work-horse, one-sector optimal growth model. Specifically, we compare RLS learning in a model that has been linearized around the steady state with learning in the original nonlinear model.

---

[7]Note that the same result in terms of E-stability would be obtained by analysing a linearized version of (3). In general though, as we will see later, this equivalence of results between nonlinear and linearized models does not hold.

4.1 The optimal growth model

The representative household seeks to maximize:

$$E_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma}$$

subject to:

$$
\begin{aligned}
c_t + k_{t+1} &\leq y_t + (1-\delta)k_t, \\
k_0 &= \hat{k}_0 \\
y_t &= z_t k_t^{\alpha} \\
z_t &= z_{t-1}^{\rho} \epsilon_t, \ z_0 = \hat{z}_0
\end{aligned}
$$

First order conditions with respect to c_t and k_{t+1} are:

$$
\begin{aligned}
c_t^{-\sigma} - \lambda_t &= 0, \\
-\lambda_t + \beta E_t \left[\lambda_{t+1} (\alpha z_{t+1} k_{t+1}^{\alpha-1} + 1 - \delta) \right] &= 0.
\end{aligned}
$$

Combining these equations and using the fact that $z_{t+1} k_{t+1}^{\alpha-1} = \frac{y_{t+1}}{k_{t+1}}$ we have:

$$c_t^{-\sigma} = \beta E_t \left[c_{t+1}^{-\sigma} \left(\alpha \frac{y_{t+1}}{k_{t+1}} + 1 - \delta \right) \right].$$

The equilibrium for this economy is thus characterized by the following set of equations:

$$
\begin{aligned}
k_{t+1} &= y_t + (1-\delta)k_t - c_t, & (7) \\
c_t^{-\sigma} &= \beta E_t \left[c_{t+1}^{-\sigma} \left(\alpha \frac{y_{t+1}}{k_{t+1}} + 1 - \delta \right) \right], & (8) \\
y_t &= z_t k_t^{\alpha}, & (9) \\
z_t &= z_{t-1}^{\rho} \epsilon_t. & (10)
\end{aligned}
$$

The model does not have a closed form solution, except in the special case of logarithmic utility function ($\sigma = 1$) and full depreciation ($\delta = 1$).[8] We will examine the performance of our algorithm in this closed-form case as well as in cases where closed-form solutions do not exist.

4.2 Learning in the linearized model

Linearizing the model around the steady state we have the following system:

$$
\begin{pmatrix} \hat{c}_t \\ \hat{k}_t \\ \hat{z}_t \end{pmatrix} = \begin{pmatrix} \phi_c & \phi_k & \phi_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} E_t \hat{c}_{t+1} \\ E_t \hat{k}_{t+1} \\ E_t \hat{z}_{t+1} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ \gamma_c & \gamma_k & \gamma_z \\ 0 & 0 & \rho \end{pmatrix} \begin{pmatrix} \hat{c}_{t-1} \\ \hat{k}_{t-1} \\ \hat{z}_{t-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \hat{\epsilon}_t \quad (11)
$$

---

[8]See Sargent (1987), p. 122.

Here, hats over variables denote deviations from steady state values. The derivation of this linearized system is shown in the Appendix (see equations (18-20). Notice that expectations matter only for the first, consumption equation in this system.

Let $x_t = (\hat{c}_t, \hat{k}_t, \hat{z}_t)$, and $v_t = \hat{\epsilon}_t$. The system can be rewritten as:

$$x_t = \phi E_t x_{t+1} + \gamma x_{t-1} + \kappa v_t.$$

where ϕ and γ are 3x3 matrices and κ is 3x1. The MSV solution takes the form

$$x_t = a + b x_{t-1} + c v_t,$$

where a, b, c are functions of the structural parameters. Analysis of learning would use PLMs of the MSV form, with unknown a, b, c:

Assume agents use PLM

$$x_t = a + b x_{t-1} + c v_t \tag{12}$$

then their forecast at time t is

$$E_t x_{t+1} = E_t \left[a + b x_t + c v_{t+1}\right] = a + ab + b^2 x_{t-1} + bc v_t$$

and the ensuing ALM

$$x_t = \phi \left[a + ab + b^2 x_t + bc v_t\right] + \gamma x_{t-1} + \kappa v_t = \phi a(1+b) + \left[\phi b^2 + \gamma\right] x_{t-1} + \left[\phi bc + \kappa\right] v_t.$$

The map from PLM to ALM gives rise to the following system of ODEs:

$$\begin{aligned} \dot{a} &= \phi a(1+b) - a \\ \dot{b} &= \phi b^2 + \phi - b \\ \dot{c} &= \phi bc + \kappa - c \end{aligned}$$

Solutions to this system of ODEs represent REE. Each REE is learnable (E-stable) iff this system of ODEs evaluated at the corresponding equilibrium is locally asymptotically stable. Therefore, E-stability is governed by the following matrices

$$\begin{aligned} \frac{d\dot{a}}{da} &= \phi(I + b) - I \\ \frac{d\dot{b}}{db} &= b' \otimes \phi + I \otimes \phi b \\ \frac{d\dot{c}}{dc} &= \phi b - I \end{aligned}$$

where I are conformable identity matrices. For E-stability to obtain, the eigenvalues of the above Jacobians, evaluated at the relevant REE, must have negative real parts. Using our baseline parameterization of the model (given in the next section) one can show, numerically, that the E-stability conditions hold.

8

E-stability is a notional-time concept. However, as Evans and Honkapohja (2001) show, under certain regularity conditions, E-stability of a rational expectations equilibrium implies stability of that same equilibrium under real-time, recursive least squares learning.

As recursive least-squares learning in the linearized model is the benchmark against which we compare our nonlinear, parameterized expectations approach, we now explain how least squares learning is implemented in the linearized model. First, suppose that agents have the perceived law of motion (12), but do not initially know the rational expectations values of those parameters. Instead, given some initial beliefs, they update their estimates of the parameters using the recursive least squares algorithm:

$$\theta_t = \theta_{t-1} + t^{-1} R_t^{-1} w_t (x_t - \theta_{t-1} w_t') \tag{13}$$

$$R_t = R_{t-1} + t^{-1} (w_t' w_t - R_{t-1}) \tag{14}$$

where $\theta = [a\ b\ c]$ is a 3x5 matrix, $w_t = [1\ x_{t-1}\ v_t]$ is 1x5, and R is a 5x5 variance covariance matrix that weighs the different elements of the vector conveying new information, giving more importance to those components that are less volatile.

As Evans and Honkapohja (2001) have pointed out, a difficulty with the use of the MSV specification $\hat{c}_t$ (12) for the optimal growth model is that it violates the assumption that the moment matrix of the regressors, $x_{t-1} = (\hat{c}_{t-1}, \hat{k}_{t-1}\ \hat{z}_{t-1})$, is positive definite. In the growth model, $\hat{c}_t$ is endogenously determined as a perfect linear combination of $\hat{k}_t$ and $\hat{z}_t$, meaning that there will be perfect collinearity among the regressors that agents use to learn. To counteract this problem, Evans and Honkapohja suggest use of a slightly modified version of PLM (12):[9]

$$\hat{c}_t = a_c + b_{ck}\hat{k}_t + b_{cz}\hat{z}_t + c_{cv}v_t, \tag{15}$$

$$\hat{k}_t = a_k + b_{kk}\hat{k}_{t-1} + b_{kz}\hat{z}_{t-1} + c_{kv}v_t,$$

$$\hat{z}_t = a_z + b_{zk}\hat{k}_{t-1} + b_{zz}\hat{z}_{t-1} + c_{zv}v_t.$$

In this re-formulation of the PLM, one uses the known, time t values of the state variables, $\hat{k}_t$ and $\hat{z}_t$ to determine $\hat{c}_t$ (the first equation of 15), while the second two equations are as before in (12) with the exclusion of consumption from the set of state variables. This change leads to a further slight change in the ensuing ALM. In our simulation analysis of learning dynamics in the linearized system, we will therefore make use of (15) rather than (12).

---

[9]An alternative approach is to add some noise to the linearized law of motion for consumption (??), e.g., due to preference shocks. However, unless this noise is sufficiently large it may not eliminate the multi-colinearity problem, and too large a noise may prevent the system from converging. For this reason we instead adopt the approach described here.

4.3 Learning in the nonlinear model: parameterized expectations in real time

We now study learning in the original nonlinear model by allowing agents to recursively estimate a parameterized function that approximates the unknown nonlinear expectational function in the Euler equation (8).

The general procedure we apply consists of the following steps:

1. Generate a series for the innovations z_t and set the initial values for $\theta's$ (we initialize values based on log-linear approximation).

2. Approximate

$$E_t g(w_{t+1}) = E_t c_{t+1}^{-\sigma} \left(\alpha z_{t+1} k_{t+1}^{\alpha-1} + 1 - \delta \right)$$

where $w_t = [c_t, z_t, k_t]$, with

$$e^{\theta' x_t}$$

where

$$
\begin{aligned}
\theta &= [\theta_0 \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]' \\
x_t &= [1 \ \log(k_t) \ \log(z_t) \ \log(k_t)^2 \ \log(z_t)^2 \ \log(k_t)\log(z_t)]'.
\end{aligned}
$$

3. Compute

$$\gamma(\theta, x_t) = e^{\theta' x_t}.$$

4. Find the implied ALM for consumption and capital:

$$
\begin{aligned}
c_t &= (\beta\gamma(\theta, x_t))^{-\frac{1}{\sigma}} \\
k_{t+1} &= z_t k_t^\alpha + (1 - \delta) k_t - c_t
\end{aligned}
$$

5. Update the $\theta's$ using the RLS procedure. First, compute $g(w_t(\theta))$, i.e., the value for g that would have implied RE on the part of agents at time $t - 1$:

$$g(w_t(\theta)) = c_t^{-\sigma} \left(\alpha z_t k_t^{\alpha-1} + 1 - \delta \right).$$

Then use the forecasting error in order to update the vector of belief parameters θ:

$$
\begin{aligned}
\theta_t &= \theta_{t-1} + t^{-1} R_{t-1}^{-1} x_{t-1} (\log(g(w_t(\theta))) - \log(\gamma(\theta, x_{t-1}))) \\
R_t &= R_{t-1} + t^{-1} (x_t x_t' - R_{t-1})
\end{aligned}
$$

6. Repeat steps 2-4 until the convergence tolerance is satisfied.

Simulations show that this learning algorithm converges; in particular, capital converges towards the (noisy) steady state value implied by the model.

4.3.1   The case $\sigma = \delta = 1$

As mentioned earlier, for the case $\sigma = \delta = 1$, a closed form solution for the model is available:

$$
\begin{aligned}
c_t &= (1 - \alpha\beta)z_t k_t^\alpha \\
k_{t+1} &= \alpha\beta z_t k_t^\alpha.
\end{aligned}
$$

We use this case to compare our approximate solution based on agents' learned parameters through our real time PE approach and the exact solution. In particular, we simulate series for consumption and capital in the two cases. As shown in Figure 1, the series generated in the two cases are very close to each other (and get closer and closer as we make the convergence criterion for the learning algorithm more tight).
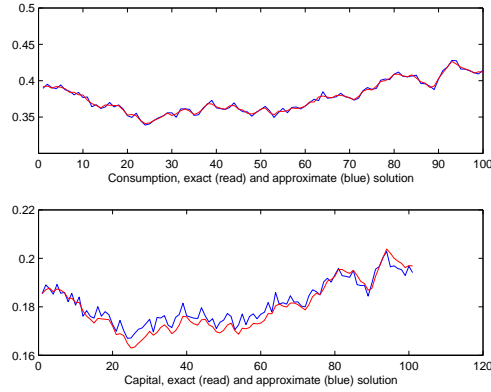

Figure 1: Approximate vs. exact solution.

4.4   A comparison

In this section we report results from a simulation study comparing real–time learning dynamics using RLS in the linearized version of the model with PE in the nonlinear version of the model. In particular, we are interested in addressing two questions.

1. Does convergence to the RE solution obtain and if so, does this depend on the parameterization of the model economy?

2. How quickly do the two algorithms converge to the RE solution?

For both learning approaches using either the linearized or the nonlinear models, we initialize the algorithm with initial beliefs that are 10% off their equilibrium values. For the linearized economy, RE

equilibrium values were obtained by solving for the minimum state variable (MSV) REE solution. For the nonlinear economy, we compute comparable "equilibrium" values using Collard's algorithm (Collard 20xx) which is based on the method proposed by Perez (2004): this basically solves the model using a log-linear approximation around the steady state, and then generates series for the endogenous variables to be used to compute values for θ.

The two economies are driven by the same realizations of the shock process. The "deep" model parameters are also held constant in comparisons between the two model economies. The baseline parameterization we used was: $\delta = .3$, $\alpha = .33$, $\rho = .95$, $\beta = .98$, $\sigma = 2$, $\sigma_\varepsilon^2 = .1$. However, we considered other model parameterizations as well following Taylor and Uhlig (1990).

We use RLS with a correctly specified model to simulate learning in the linearized model, while we use the RLS PE approach outlined above to simulate learning in the nonlinear model. In this case, agents learn an approximate version of the RE equilibrium. In both cases we assume agents know the process for productivity: as this is an exogenous stochastic process, it could otherwise be easily estimated with least squares techniques, which would be consistent and efficient. Agents therefore will need to learn 2 equations (6 parameters) in the linear case and 1 equation (6 parameters) in the non linear case.

We conducted 100 simulations of the two learning model specifications for each of 7 different sets of model parameters (these parameter sets are given in Table 1) and we recorded whether convergence occurred according to our criterion and if so, the average and variance in the number of periods it took for each algorithm to converge to the RE solution. The tolerance criterion (size of the larger update in the parameter estimates) is set to 1e-5 for both algorithms.

In response to the first question we posed, we can report that for all simulation runs and for all parameter sets we considered, both algorithms *always converged* to the RE solution. This finding indicates that, in terms of convergence, the RLS PE approach is no worse than the RLS approach using the linearized model, and that both do well under a variety of different parameterizations of the model that researchers have found interesting. Indeed, as the real-time PE approach does not require linearization of the of the model in advance and nevertheless converges to (an approximation of) the nonlinear RE solution, we can argue that this is a more reasonable way of characterizing how agents might go about learning that also provides a microfoundation for the use of the rational expectations hypothesis.

In response to the second question we posed, Table 1 reports on the average (and variance) in the number of periods required before convergence was declared for each of the 7 sets of model parameters we considered. From this Table we have the following findings

1. The average time of convergence is almost always greater for the PE nonlinear algorithm as compared with the RLS algorithm using the linearized model. The one exception occurs for the case where σ is changed

from the baseline value of 2 to 0.50.

2. A lower δ reduces time of convergence for the nonlinear case but has a non-monotonic impact on the linear case;

3. A higher α (.5) reduces time of convergence both for the linear and nonlinear case;

4. A lower σ (.5) reduces time of convergence for both linear and nonlinear case (but more so for the nonlinear case);

5. The time of convergence for both the linear and nonlinear cases depends on σ_ε^2. A lower σ_ε^2 decreases the time of convergence.

Model Parameterization[a]	RLS Linearized case		RLS-PE Nonlinear case	
	Avg Time	Variance	Avg Time	Variance
Baseline Case	353.19	3.8661×10^4	634.87	6.3407×10^5
$\delta = 0.2$	383.59	4.8987×10^4	459.11	1.2741×10^5
$\delta = 0.1$	327.72	2.5536×10^4	376.39	6.2811×10^4
$\alpha = 0.5$	337.06	4.2785×10^4	623.16	6.6828×10^5
$\sigma = 1.0$	315.46	3.3360×10^4	302.92	3.7057×10^5
$\sigma = 0.5$	254.52	2.1698×10^4	79.85	2.4446×10^3
$\sigma_\varepsilon^2 = .02$	225.84	1.6966×10^4	258.85	2.2684×10^4

a. Parameterizations other than the baseline case differ from the baseline case only in the one parameter choice indicated.

Table 1: Average Time to Convergence and Variance from 100 Simulations of the RLS-Linearized and RLS-PE-Nonlinear Models Under Various Model Parameterizations

For the special case $\delta = \sigma = 1$, the learning algorithm becomes very unstable for the linear case, and it does not converge. In the nonlinear case, instead, this parameterization does not seem to create any problems, and in fact the learning algorithm converges, on average, in 74.79 periods.

These results show that, for the optimal growth model, the nonlinear, parameterized expectations approach can be implemented in real time without much loss in terms of speed of convergence and regions of the parameter space where convergence obtains relative to the linearized model. Moreover, changing the value of some parameters (for example, δ) can have different effects in terms of speed of convergence for the linear and nonlinear case.

These findings suggest that learning an approximate version of the equilibrium in a nonlinear model might not be any more complicated, in terms of computational requirements, than learning in a linear(ized) setting, and in fact it requires less information, as it does not require agents to know the steady state of the model around which the linearization is carried out.

5 Conclusions

The stability of rational expectations equilibrium under adaptive learning dynamics has been the subject of a large literature (Evans and Honkapohja (2001)) and provides a important micro-foundation for the rational expectations hypothesis. However, much of this literature presumes that agents use a correct, reduced-form, linear specification of the rational expectations model when forming expectations of future endogenous variables. For this approach to make sense, the researcher must first solve for the steady state of the model and then linearize the model around that steady state solution. It is at this point that the researcher replaces rational expectations in the linearized system with adaptive expectations formed by learning agents, whose expectations follow from their use of a linear perceived law of motion that is consistent with the solution to the linearized model. This procedure implies that learning agents begin their adaptive process of learning with tremendous prior knowledge of the system in which they operate.

Our approach is to relax the assumption that agents use a linear, perceived law of motion and instead suppose that they form forecasts using a nonlinear, parameterized expectations approach. While the parameterized expectations method was originally devised as a means by which researchers could find rational expectations solutions in complicated nonlinear dynamic, stochastic models, we show how the updating procedure for PE can be accomplished in *real-time*, using new observations generated by the interaction between the parameterized expectations and the actual nonlinear stochastic, dynamical system, so that PE might be viewed as a rival to other real-time learning algorithms used in linearized systems such as recursive least squares.

Our main finding is that the real-time parameterized expectations approach involving an approximation of nonlinear expectations of future variables always converges to the rational expectations equilibrium of the optimal growth model that we study under a variety of different parameterizations of that model. Further, we find that convergence under this nonlinear learning approach is not much slower than under least squares learning using the linearized version of the same model, and is even, under some parameterizations, faster. These results suggest that nonlinear real-time learning models can serve as a useful and more plausible alternative to linear learning models particularly in environments where the system that agents are seeking to learn is nonlinear. The application we have considered is a nonlinear system with a unique rational expectations equilibrium. It is still unknown how well our real-time PE approach would work in models with multiple rational expectations equilibria. We leave that exercise to future research.

Appendix

Linearization of the growth model

Linearization proceeds as follows. First define the variable $\hat{x}_t = \ln\left(\frac{x_t}{\bar{x}}\right)$, where $\bar{x}$ is the steady state value of variable x, i.e., $\hat{x}_t$ denotes the deviation from the steady state value. Note further from Uhlig (1999) that $x_t = \bar{x}e^{\hat{x}_t}$ and that $e^{x_t} \approx 1 + x_t$. For equation (7) we have:

$$\bar{k}e^{\hat{k}_{t+1}} = \bar{y}e^{\hat{y}_t} + (1-\delta)\bar{k}e^{\hat{k}_t} - \bar{c}e^{\hat{c}_t} \text{ or}$$

$$\bar{k}(1+\hat{k}_{t+1}) = \bar{y}(1+\hat{y}_t) + (1-\delta)\bar{k}(1+\hat{k}_t) - \bar{c}(1+\hat{c}_t)$$

Using the fact that $\bar{y} = \bar{c} + \delta\bar{k}$, we can simplify the above expression as:

$$\hat{k}_{t+1} = \frac{\bar{y}}{\bar{k}}\hat{y}_t + (1-\delta)\hat{k}_t - \frac{\bar{c}}{\bar{k}}\hat{c}_t \tag{16}$$

For equation (9) we can write:

$$\bar{y}e^{\hat{y}_t} = \bar{z}\bar{k}^\alpha e^{\hat{z}_t + \alpha\hat{k}_t}$$

$$\approx \bar{y}(1+\hat{y}_t) = \bar{z}\bar{k}^\alpha(1+\hat{z}_t + \alpha\hat{k}_t).$$

Since in a steady state $\bar{y} = \bar{z}\bar{k}^\alpha$, we have:

$$\hat{y}_t = \hat{z}_t + \alpha\hat{k}_t. \tag{17}$$

Substituting (17) into (16) we have:

$$\hat{k}_{t+1} = \gamma_z \hat{z}_t + \gamma_k \hat{k}_t + \gamma_c \hat{c}_t, \tag{18}$$

where $\gamma_z = \frac{\bar{y}}{\bar{k}}$, $\gamma_k = \alpha\gamma_z + 1 - \delta$ and $\gamma_c = -\frac{\bar{c}}{\bar{k}}$

We can be more precise about the γ coefficient values as it is straightforward to calculate $\frac{\bar{y}}{\bar{k}}$ and $\frac{\bar{c}}{\bar{k}}$. From (9), we have that in a steady state:

$$\frac{\bar{y}}{\bar{k}} = \frac{1-\beta(1-\delta)}{\alpha\beta}$$

From (8) we have that, in a steady state:

$$\frac{\bar{c}}{\bar{k}} = \frac{\bar{y}}{\bar{k}} - \delta = \frac{1-\beta+\beta\delta(1-\alpha)}{\alpha\beta}$$

It follows that $\gamma_z = \frac{1-\beta(1-\delta)}{\alpha\beta}$, $\gamma_k = \frac{1}{\beta}$ and $\gamma_c = \frac{-1+\beta-\beta\delta(1-\alpha)}{\alpha\beta}$.

For equation (8) we have:

$$\bar{c}^{-\sigma}e^{-\sigma\hat{c}_t} = \beta E_t\left[\bar{c}^{-\sigma}e^{-\sigma\hat{c}_{t+1}}\left(\alpha\frac{\bar{y}e^{\hat{y}_{t+1}}}{\bar{k}e^{\hat{k}_{t+1}}} + 1 - \delta\right)\right]$$

$$\approx (1-\sigma c_t) = \beta\left[(1-\sigma E_t\hat{c}_{t+1})\left(\alpha\frac{\bar{y}(1+E_t\hat{y}_{t+1})}{\bar{k}(1+E_t\hat{k}_{t+1})} + 1 - \delta\right)\right]$$

or,

$$1 = \beta \left[\frac{1 - \sigma E_t \hat{c}_{t+1}}{1 - \sigma \hat{c}_t} \left(\alpha \frac{\bar{y}(1 + E_t \hat{y}_{t+1})}{\bar{k}(1 + E_t \hat{k}_{t+1})} + 1 - \delta \right) \right]$$

In a steady state, we have that $\beta(1 - \delta) = 1 - \beta \alpha \frac{\bar{y}}{\bar{k}}$ Therefore, we can rewrite the above as:

$$1 = \beta \alpha \frac{1 - \sigma E_t \hat{c}_{t+1}}{1 - \sigma \hat{c}_t} \left(\frac{\bar{y}(1 + E_t \hat{y}_{t+1})}{\bar{k}(1 + E_t \hat{k}_{t+1})} \right) + \frac{1 - \sigma E_t \hat{c}_{t+1}}{1 - \sigma \hat{c}_t} (1 - \beta \alpha \frac{\bar{y}}{\bar{k}}),$$

or

$$1 = \beta \alpha \frac{1 - \sigma E_t \hat{c}_{t+1}}{1 - \sigma \hat{c}_t} \left(\frac{\bar{y}(1 + E_t \hat{y}_{t+1})}{\bar{k}(1 + E_t \hat{k}_{t+1})} \right) + \frac{(1 - \sigma E_t \hat{c}_{t+1})(1 + E_t \hat{k}_{t+1})}{(1 - \sigma \hat{c}_t)(1 + E_t \hat{k}_{t+1})} (1 - \beta \alpha \frac{\bar{y}}{\bar{k}})$$

or

$$(1 - \sigma \hat{c}_t)(1 + E_t \hat{k}_{t+1}) = \beta \alpha \frac{\bar{y}}{\bar{k}} (1 - \sigma E_t \hat{c}_{t+1})(1 + E_t \hat{y}_{t+1}) + (1 - \sigma E_t \hat{c}_{t+1})(1 + E_t \hat{k}_{t+1})(1 - \beta \alpha \frac{\bar{y}}{\bar{k}}).$$

Using the approximation that $\hat{x}\hat{y} \approx 0$, the above simplifies to:

$$1 - \sigma \hat{c}_t + E_t \hat{k}_{t+1} = \beta \alpha \frac{\bar{y}}{\bar{k}} (1 - \sigma E_t \hat{c}_{t+1} + E_t \hat{y}_{t+1}) + 1 - \sigma E_t \hat{c}_{t+1} + E_t \hat{k}_{t+1} - \beta \alpha \frac{\bar{y}}{\bar{k}} (1 - \sigma E_t \hat{c}_{t+1} + E_t \hat{k}_{t+1}),$$

or

$$-\sigma \hat{c}_t = \beta \alpha \frac{\bar{y}}{\bar{k}} E_t \hat{y}_{t+1} - \sigma E_t \hat{c}_{t+1} - \beta \alpha \frac{\bar{y}}{\bar{k}} E_t \hat{k}_{t+1}.$$

Using the definition of $\hat{y}_t$ (17) we can write:

$$\hat{c}_t = \phi_z E_t \hat{z}_{t+1} + \phi_k E_t \hat{k}_{t+1} + \phi_c E_t \hat{c}_{t+1}, \tag{19}$$

where $\phi_z = -\frac{\alpha \beta}{\sigma} \frac{\bar{y}}{\bar{k}}$, $\phi_k = (\alpha - 1)\phi_z$ and $\phi_c = 1$. Using the fact that $\frac{\bar{y}}{\bar{k}} = \frac{1 - \beta(1-\delta)}{\alpha\beta}$, we have that $\phi_z = (\beta(1 - \delta) - 1)/\sigma$ and $\phi_k = (\alpha - 1)(\beta(1 - \delta) - 1)/\sigma$.

Finally, equation (10) can be written as:

$$\bar{z} e^{\hat{z}_t} = \rho \bar{z} e^{\hat{z}_{t-1}} + \hat{\epsilon}_t$$

$$\approx \bar{z}(1 + \hat{z}_t) = \rho \bar{z}(1 + \hat{z}_{t-1}) + \hat{\epsilon}_t,$$

or

$$\hat{z}_t = \rho \hat{z}_{t-1} + \hat{\epsilon}_t \tag{20}$$

References

[1] Chen, X. and White, H. (1998). Nonparametric adaptive learning with feedback. *Journal of Economic Theory* 82, 190-222.

[2] Collard, F. (). Parameterized expectations algorithm. Mimeo (available at http://fabcol.free.fr/).

[3] Den Hann, W.J., Marcet, A. (1990). Solving the stochastic growth model by parameterizing expectations. *Journal of Business and Economic Statistics* 8, 31-34.

[4] Eusepi, S. (2007). Learnability and monetary policy: a global perspective. *Journal of Monetary Economics* 54, 1115-1131.

[5] Evans, G.W., Honkapohja, S. (1995). Local convergence of recursive learning to steady states and cycles in stochastic nonlinear models. *Econometrica* 63, 195-206.

[6] Evans, G.W., Honkapohja, S. (2001). *Learning and Expectations in Macroeconomics*. Princeton University Press.

[7] Evans, G.W., Honkapohja, S. (2006). Monetary policy, expectations and commitment. *Scandinavian Journal of Economics* 108, 15–38.

[8] Evans, G.W., Honkapohja, S. and Mitra, K. (2003). Notes on agents' behavioral rules under adaptive learning and recent studies of monetary policy. Working paper, University of Oregon.

[9] Evans, G.W. and McGough, B. (2009) Learning to optimize. Working paper, University of Oregon.

[10] Marcet, A. (1988). Solving non-linear stochastic models by parameterized expectations. Working paper, Carnegie Mellon University.

[11] Marcet, A., Lorenzoni, G. (1999). Parameterized expectations approach: some practical issues. Working paper, Universitat Pompeu Fabra. Also in: Marimon, R., Scott, A. (editors), *Computational methods for the study of dynamic economies*, Oxford University Press, 1999, 143-171.

[12] Marcet, A., Marshall, D.A. (1994). Solving nonlinear rational expectations models by parameterized expectations: convergence to stationary solutions. Federal Reserve Bank of Minneapolis Discussion paper 91.

[13] Perez, J.J. (2004). A log-linear homotopy approach to initialize the parameterized expectations algorithm. *Computational Economics* 24, 59–75.

[14] Preston, B. (2005). Learning about monetary policy rules when long-horizon forecasts matter. International Journal of Central Banking, 1.

[15] Sargent, T.J. (1987). Dynamic Macroeconomic Theory, Harvard University Press.

[16] Taylor, J.B., Uhlig, H. (1990). Solving nonlinear stochastic growth models; a comparison of alternative solution methods. *Journal of Business and Economics Statistics* 8, 1-18.

[17] Uhlig, H. (1999). A Toolkit for Analyzing Nonlinear Dynamic Stochastic Models Easily, in: Marimon, R., Scott, A. (editors), *Computational methods for the study of dynamic economies*, Oxford University Press, 1999, 30-61.