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

Revised Draft
September 2012

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 argue that 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 and we provide a comparison of the two approaches

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

*We thank an associate editor, two referees and participants in the IHS Workshop in Honor of Stephen J. Turnovsky for helpful comments on earlier drafts of this paper. Berardi gratefully acknowledges financial support from the European Community's Seventh Framework Programme (FP7/2007-2013), grant # 225408 (POLHIA).

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 (i.e., 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 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 is conditionally linear and so local stability can be assessed using standard analytic techniques. On the other hand, there is no reason to presume that adaptive, learning agents will use linear forecast rules, especially in the often non-linear dynamic recursive stochastic models in which these learning agents are situated. In this paper we explore the possibility that learning agents use nonlinear forecasting rules and that they update the coefficients of their nonlinear rules in real time. An advantage of our approach is that it is no longer necessary to first linearize the REE model around its steady state as the nonlinear forecasts can be included in the original nonlinear model in place of rational expectations forecasts. Furthermore, stability analysis under this nonlinear learning rule dynamic may be interpreted as providing a more global stability analysis, as opposed to the local analysis that necessarily follows from using linearized versions of the REE model. A disadvantage of our approach is that results on the stability of the nonlinear learning system and related findings such as speed of convergence will necessarily be numerical in nature.

Specifically, we suppose that learning agents adopt the method of parameterized expectations that Marcet (1988) introduced as a method for 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. Marcet (1988) and Marcet and Marshall (1994) proposed that this method could be useful not only for finding rational expectations solutions but also as a model of real-time adaptive learning behavior. However they did not pursue the latter application of parameterized expectations. In this paper we follow up on their suggestion and consider the performance of the parameterized expectations algorithm as a real-time model of adaptive learning.

More precisely, the main contribution of this paper is to demonstrate the feasibility of real-time, nonlinear, parameterized expectations learning and to compare and contrast the use of this approach with the more standard approach of least squares learning that involves working with dynamical systems that have been linearized around steady state values. We are not aware of any prior comparison of these two different real-time learning approaches in terms of whether convergence occurs and, if so, how quickly convergence takes place. Indeed our aim here is to assess whether and how the method of real-time adaptive, nonlinear, parameterized expectations learning differs from least squares learning in the context of a standard dynamic stochastic general equilibrium macroeconomic system and, in the process we also address some important implementation issues such as the choice of the convergence criterion, the specification of the nonlinear forecast rule and the initialization of our real-time algorithm. If performance differences (e.g., in speed of convergence) are not too great and if implementation is not too problematic then it may be reasonable to prefer the real-time nonlinear, parameterized expectations learning approach over linear least squares learning, as the former can be given a "global" as opposed to a "local" stability interpretation.

Our approach to learning lies within a framework that Evans and Honkapohja (2006) have termed "Euler-equation" learning as the agents in our model take the Euler equation as the fundamental primitive and 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 approaches have been used to examine stability under learning in *linearized* versions of macroeconomic systems.

We note that Evans et al. (2007) have considered the case where agents form expectations of the nonlinear, right-hand-side expression in their consumption Euler equation in a cash-in-advance macroeconomic model using a simple, state contingent linear past-averaging approach that requires information only about the current state. This expectation is then incorporated into the nonlinear equations of the macroeconomic model. Simulations are used to confirm the convergence of this learning system to a Markov stationary sunspot equilibrium. By contrast, we suppose that agents's learning rule is itself a nonlinear function of current state variables and this forecast is then embedded in the nonlinear data generating equations of the macroeconomic system in which agents are learning. 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 agents in their model forecast one-step ahead values for shadow prices, which they then use to make current period decisions.

2 The Method of Parameterized Expectations

The method of parameterized expectations was proposed by Marcet (1988) as a way to *solve* for the RE equilibrium solution in nonlinear models, by approximating an unknown nonlinear expectational function with a parametric function.[1] The general form of the model to be solved 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.

The general strategy is to approximate the expectational function, $E_t[g(z_{t+1}, z_t)]$, with a flexible parametric function $\gamma(\theta, x_t)$,[2] where θ is the vector of parameters and x_t is a 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

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

A common approach is to start with some initial guess, θ_0 (and some initial values, x_0), and to use $\gamma(\theta_0, x_t)$ to generate a long series of data, $\{z_t\}_{t=1}^T$. One then uses this data to run the nonlinear least squares regression:

$$g(z_{t+1}, z_t) = \gamma(\theta_0, x_t) + u_t,$$

to obtain an estimate, $\hat{\theta}$. Finally one uses this estimate to update the guess for θ:

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

[1] 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).

[2] 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.

where $\phi \in (0, 1)$ is a smoothing parameter. This process is applied iteratively in this same fashion until a convergence criterion is met.

While the method of PE was originally developed as an off–line algorithm to be used to find the REE solution for a model, it is possible to re-interpret it as an on–line, or "real–time" algorithm used by adaptive agents in the model to learn the structure of the economy. In the *real-time* interpretation, the realization of the parameter vector of the nonlinear approximating function at date $t-1, \theta_{t-1}$, enters into the actual laws of motion determining realizations of the date t endogenous variable(s) that agents are learning. By contrast, in the off-line interpretation, the researcher uses iterations of the nonlinear regression to find a good approximation to the nonlinear expectational function; once that function is obtained (after the convergence criterion has been satisfied), then (and only then) the approximating expectational function is used to determine realizations of the associated endogenous variable(s) in simulations of the model. Indeed, as mentioned in the introduction, Marcet and Marshall (1994) suggested that a real-time implementation of the PE approach might be possible, but they did not pursue it.[3] We therefore investigate the performance of the PE algorithm as a recursive real-time learning algorithm when the equations agents need to learn are nonlinear (and, to the agents, 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

Learning in nonlinear models has been the focus of a small literature. One approach that has been taken is to posit that agents mistakenly use linear forecast rules in non-linear models, see, e.g., Hommes and Sorger (1998), Lansing (2006) and Bullard et al. (2010) among others. Our encompassing approach allows agents to adopt such linear forecast rules, but

[3] Marcet and Marshall (1994, appendix 2) provide conditions, based on the stochastic approximation theory of Ljung (1975), that would ensure *local* stability of the non-linear least squares algorithm to rational expecations equilibrium but do not implement this algorithm or compare its performance with recursive least squares learning in the linearized system as we do in this paper.

it also allows them to discover the relevant nonlinear forecast rules for the nonlinear model economy. A second approach to learning in nonlinear models focus on steady states or k-cycles of the (stationary) nonlinear models and thus avoids the need to find an approximate solution for the expectational function, see, e.g., Evans & Honkapohja (1995) and (2001, chapters 11 and 12), and Eusepi (2007). Specifically, given the nonlinear model

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

these papers 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 mod $k = i$ in case of a k-cycle ($k = 1$ in case of a steady state). As agents are assumed to know in advance whether 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. (e.g., there are no lagged components in the structural equations, and the shocks are i.i.d. 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 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 approximating function F.[5]

[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.

4 An application to the optimal growth model

We illustrate our approach of using real-time learning by means of parameterized expectations by applying it to the work-horse, one-sector optimal growth model. Specifically, we compare recursive least squares learning in a model that has been linearized around the steady state with our approach of real-time parameterized expectations learning in the original nonlinear model.

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:

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

Here, c_t, k_t and y_t denote time t consumption, capital and output respectively; z_t is a technology shock with innovation $\epsilon_t \sim N(0, \sigma_\epsilon^2)$. Other model parameters include the period discount factor $\beta \in (0, 1]$, the coefficient of relative risk aversion, $\sigma > 0$, the depreciation rate of capital $\delta \in (0, 1]$,and capital's share of output $\alpha \in (0, 1)$.

The 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:

$$
\begin{aligned}
k_{t+1} &= y_t + (1-\delta)k_t - c_t, &(3)\\
c_t^{-\sigma} &= \beta E_t \left[c_{t+1}^{-\sigma} \left(\alpha \frac{y_{t+1}}{k_{t+1}} + 1 - \delta \right) \right], &(4)\\
y_t &= z_t k_t^{\alpha}, &(5)\\
z_t &= z_{t-1}^{\rho} \epsilon_t. &(6)
\end{aligned}
$$

The model does not have a closed form solution, except in the special case of logarithmic utility function ($\sigma = 1$) and full depreciation ($\delta = 1$).[6] 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.

While our real-time parameterized expectation model will use the model described by equations (3)-(6), the standard real-time recursive least squares learning approach requires that we first linearize the model around the steady state. Performing this linearization results in the following system:

$$
\begin{pmatrix} \hat{c}_t \\ \hat{k}_t \\ \hat{z}_t \end{pmatrix} = \begin{pmatrix} \phi_c & \phi_k & \phi_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} E_t \hat{c}_{t+1} \\ E_t \hat{k}_{t+1} \\ E_t \hat{z}_{t+1} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ \gamma_c & \gamma_k & \gamma_z \\ 0 & 0 & \rho \end{pmatrix} \begin{pmatrix} \hat{c}_{t-1} \\ \hat{k}_{t-1} \\ \hat{z}_{t-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \hat{\epsilon}_t. \quad (7)
$$

Here, hats over variables denote deviations from steady state values. The derivation and coefficient values for this linearized system are described briefly in the Appendix (see equations (14-16).

4.2 Learning in the linearized model: recursive least-squares learning

Let $x_t' = (\hat{c}_t, \hat{k}_t, \hat{z}_t)$, and $v_t = \hat{\epsilon}_t$. The system (7) can be rewritten more compactly as:

$$
x_t = \phi E_t x_{t+1} + \gamma x_{t-1} + \kappa v_t. \quad (8)
$$

where ϕ and γ are 3×3 matrices and κ is 3×1.

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

A standard adaptive learning approach for linearized multivariate systems of the form (8) is to focus on the stability of the minimal state variable (MSV) RE solution under recursive least squares (RLS) learning as described in Evans and Honkapohja (2001, Chapter 10)). Given the popularity of this method, we view it as an important benchmark for our real-time nonlinear PE learning approach.

The MSV RE solution is of the form:

$$x_t = a + bx_{t-1} + v_t. \tag{9}$$

Under rational expectations, a is 3×1 vector and b is a 3×3 matrix having elements that are functions of structural model parameters. However, under the assumption that agents are adaptive learners, one relaxes the RE assumption and supposes instead that learning agents do not initially know the true RE values of a and b, (though they do employ the correct linear MSV specification). Instead, using some other parameterization for $(a,\, b)$ they use 9 as a perceived law of motion (PLM) to forecast future variables:

$$E_t x_{t+1} = E_t \left[a + bx_t\right] = (I + b)a + b^2 x_{t-1}.$$

where I denotes a 3×3 identity matrix. Substituting this forecast into the system (8), we obtain the "actual" law of motion (ALM) for the economy under adaptive learning:

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

The mapping from the PLM to the ALM is given by:

$$T(a, b) = (\phi(I + b)a, \phi b^2 + \gamma),$$

and the fixed points of this mapping comprise the MSV RE solution of the system.

Stability of this system under adaptive learning can be assessed using the techniques of Evans and Honkapohja (2001). In particular, using the rules for vectorization of matrix products, we calculate:

$$\begin{aligned} DT_a(\bar{b}) &= \phi(I + \bar{b}), \\ DT_b(\bar{b}) &= \bar{b}' \otimes \phi + I \otimes \phi\bar{b}. \end{aligned}$$

where $\bar{b}$ denotes the steady state, REE value, i.e., the fixed point of the T-mapping for b. The REE is learnable (E-stable) if all eigenvalues of the matrices $DT_a(\bar{b})$ and $DT_b(\bar{b})$ have real parts less than one. Using our baseline parameterization of the model (given in the next section) one can show, numerically, that these E-stability conditions hold.

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 (RLS) learning.

As RLS learning in the linearized model is the benchmark against which we compare our nonlinear, parameterized expectations approach, we now explain how RLS learning is implemented in the linearized model. First, suppose that agents have the perceived law of motion (9), 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 RLS algorithm:

$$\theta_t = \theta_{t-1} + t^{-1} R_t^{-1} w_t (x_t - \theta_{t-1} w_t'), \tag{10}$$

$$R_t = R_{t-1} + t^{-1}(w_t' w_t - R_{t-1}), \tag{11}$$

where $\theta = [a\ b]$ is a 3×4 coefficient matrix, $w_t = [1\ x_{t-1}]$ is a 1×4 matrix of regressor variables and R is a 4×4 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 $\hat{c}_t$ (9) 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 (9):[7]

$$\begin{aligned} \hat{c}_t &= a_c + b_{ck}\hat{k}_t + b_{cz}\hat{z}_t + v_t, \\ \hat{k}_t &= a_k + b_{kk}\hat{k}_{t-1} + b_{kz}\hat{z}_{t-1} + v_t, \\ \hat{z}_t &= a_z + b_{zz}\hat{z}_{t-1} + v_t. \end{aligned} \tag{12}$$

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 12), while the second two equations are as in (9) but exclude consumption from the set of state variables. This change leads to a further slight change in the ensuing ALM as well. In our simulation analysis of learning dynamics in the linearized system, we will therefore make use of (12) rather than (9). Alternatively we will also consider the case where agents are presumed to know the laws of motion for k_t and z_t and are only learning about consumption, c_t, which is, after all, the only variable for which future expectations play a role.

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 (4).

The general procedure we apply consists of the following steps:

1. Generate a series for the innovations $\{z_t\}$ and set the initial parameter values in the vector θ.

2. Approximate the expectational function,

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

[7]An alternative approach is to add some noise to the linearized law of motion for consumption in (7), e.g., due to preference shocks. However, unless this noise is sufficiently large it may not eliminate the multicolinearity problem, and too large a noise may prevent the system from converging. For this reason we instead adopt the approach described here.

where $w_t' = [c_t, z_t, k_t]$, using a polynomial function. Following the parameterized expectations literature we use an *exponential* polynomial function to approximate the nonlinear right hand side of the Euler equation:

$$e^{\theta_{t-1}' x_t},$$

where

$$\begin{aligned} \theta_{t-1}' &= [\theta_0 \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5], \\ x_t' &= [1 \ \log(k_t) \ \log(z_t) \ \log(k_t)^2 \ \log(z_t)^2 \ \log(k_t)\log(z_t)]. \end{aligned} \tag{13}$$

Thus, we use a *second*-order polynomial in our exponential approximation function. This is the minimal order polynomial specification that allows agents to approximate the nonlinear RE solution to the model. Depending on the application, higher than second-order polynomial approximations might be appropriate.[8] We note that since we are using a finite-order polynomial approximation, the solution that subjects may learn using this specification can only approximate the rational expectations solution. Thus, one might prefer to say that agents in our framework are learning an "approximate" "perturbed" or, in the language of Evans and Honkapohja (2001), a "restricted perceptions" equilibrium, rather than the true rational expectations solution, as there is some mis-specification in the form of the forecast rule that they adopt and this mean that the stationary solution to which the system converges may only approximate the RE solution.

3. Compute

$$\gamma(\theta_{t-1}, x_t) = e^{\theta_{t-1}' x_t}.$$

4. Find the implied ALM for consumption and capital:

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

[8] We will address the question of the order of the polynomial approximation later in section 5.2, where we will consider the case where learning agents use a third-order polynomial approximation.

5. Update θ_{t-1}using the RLS procedure. First, compute $g(w_t(\theta))$:

$$\begin{aligned} g(w_t(\theta_{t-1})) &= c_t^{-\sigma}\left(\alpha z_t k_t^{\alpha-1}+1-\delta\right) \\ &= \beta\gamma(\theta_{t-1},x_t)\left(\alpha z_t k_t^{\alpha-1}+1-\delta\right). \end{aligned}$$

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

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

6. Repeat steps 2-5 until a convergence criterion is satisfied. We use the criterion that $\max_i|\theta_t^i-\theta_{t-1}^i|<\tau$ is satisfied, where i indexes the parameters of approximation function (13). This convergence criterion is *operational* in the sense that learning agents do not have to known (and, of course would not know) the true rational expectations parameter vector, θ^*, in order to declare an end to their attempted learning of that solution. All they need to evaluate is the maximum parameter value difference (in absolute terms) from one period to the next relative to some chosen tolerance, τ.

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

We start with the special case $\sigma=\delta=1$, where a closed form solution for the nonlinear model is available and given by:

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

We use this case to compare our approximate solution based on agents' learned parameter estimates as obtained from our real time PE approach with the exact solution. In particular, we simulate a series for consumption and capital for these two cases (real-time estimated and exact solutions). As shown in Figure 1, the series generated in the two cases are very close to each other. This comparison provides us with some assurance that our real-time PE algorithm can converge to an approximation of the nonlinear RE solution.

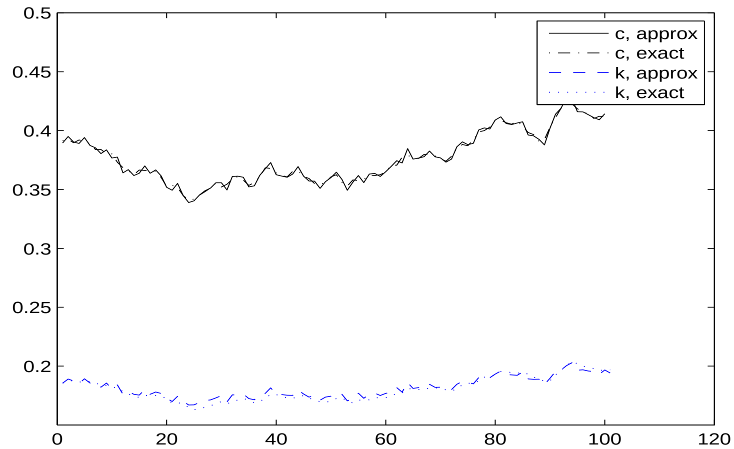

Figure 1: Consumption and capital series based on estimates from the the real time PE approach compared with series generated by the exact solution.

4.4 A comparison

In the next section 5 we report results from a simulation study comparing real–time learning dynamics using RLS in the linearized version of the model with real-time PE learning in the nonlinear version of the model. In conducting this comparison we seek to demonstrate the feasibility of the real-time, nonlinear PE learning approach by addressing the following two main questions:

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

2. How quickly does each algorithm converge to the RE solution?

In addition to answering these two main questions, we will also address some further robustness and implementation issues.

For both learning approaches using either the linearized or the nonlinear models, we initialized the algorithm with initial beliefs that are 10% off (i.e., higher than) their respective RE equilibrium values. For the linearized economy, the RE equilibrium parameter values,

θ^*, were obtained by solving for the minimum state variable (MSV) REE solution. For the nonlinear economy, we compute comparable RE equilibrium values using Collard's parameterized expectations (PE) algorithm (Collard 2002) which is based on the method proposed by Perez (2004): this approach solves the model using a log-linear approximation around the steady state, and then generates a series for the endogenous variables that enables us to compute values for θ^*.

To make the comparison as clean as possible, the two model economies (linear and nonlinear) are driven by the same realizations for the shock process. Further, the "deep" structural parameters of the model are held constant in comparisons between the two learning algorithms. The baseline model parameterization we adopted (assuming a quarterly frequency) was: $\alpha = .33$, $\delta = .3$, $\rho = .95$, $\beta = .98$, $\sigma = 2$, $\sigma_\varepsilon^2 = .1$.[9] However, we also consider a number of other model parameterizations that have been used for this model as proposed by 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 the latter case, recall again that agents learn an approximate version of the RE equilibrium. Note also that agents have to learn a different number of equations (and parameters) in the two cases: 3 equations and 8 parameters in the linear, RLS case (the parameters of the PLM (12) and 1 equation (6 parameters) in the non-linear PE case (the parameters of the polynomial approximation function, (13).[10]

We conducted 100 simulations of the two learning approaches for each of 9 different sets of model parameters which are given in Table 1. For each run, we recorded whether convergence occurred according to our operational criterion, $\max_i\{|\theta_t^i - \theta_{t-1}^i|\} < \tau$, where

[9] Our baseline depreciation rate $\delta = .3$ may be regarded as high for a model with a quarterly frequency of observation. We made this choice because more empirically plausible values for δ led to a sometimes unstable learning algorithm. Nevertheless, we report results in Table 1 for the empirically more plausible (and standard) parameterization where $\delta = .025$.

[10] As we do not assume any cost of learning, this difference does not create any problems. Nevertheless, we will later consider cases where the linear RLS learning algorithm involves less than 8 parameters to be learned.

i indexes the number of parameters to be learned. If convergence obtains we record the average and variance in the number of periods, t, it took for each algorithm to satisfy that convergence criterion. The tolerance level, τ, was set equal to $1e-5$ for both algorithms and the maximum number of periods allowed for each simulation run was set at 100,000.

5 Findings

5.1 Convergence

In response to the first question we posed, we can report that for all 9 parameter sets we considered, both algorithms converge to the RE solution according to our convergence criterion and well within our upper bound of 100,000 periods –see Table 1 for mean convergence times and variances over the 100 runs of the 9 model parameterizations. This finding indicates that, n terms of convergence alone, the Real Time-PE approach provides a simple and tractable way in which we can allow for nonlinearities in expectation formation. Thus, our nonlinear, real-time PE approach may be considered an alternative the use of linearization methods to approximate expectations and RLS to model the updating of those linearized forecast rules.

Figure 2 shows the evolution of 3 parameter estimates for the consumption equation in (12) for the linearized model as obtained from a single, representative run of the RLS algorithm using our baseline parameterization. While the convergence criterion is satisfied in this case after several hundred iterations, we show the parameter estimates over the longer horizon of 100,000 periods.

This figure reveals that there is still some very slow adjustment that is taking place in these RLS parameter estimates even toward the limit of 100,000 observations. Specifically, the precise MSV rational expectation solution values for the parameters of the consumption equation are $\theta^* = [0, .3643, .8403]$, while the estimated parameter values from the run of the RLS algorithm depicted in Figure 2 at period 100,000 is: $\theta_{t=100,000} = [-.0267, .3693, .8806]$. Thus, while convergence is satisfied according to our convergence criterion, there is a sense

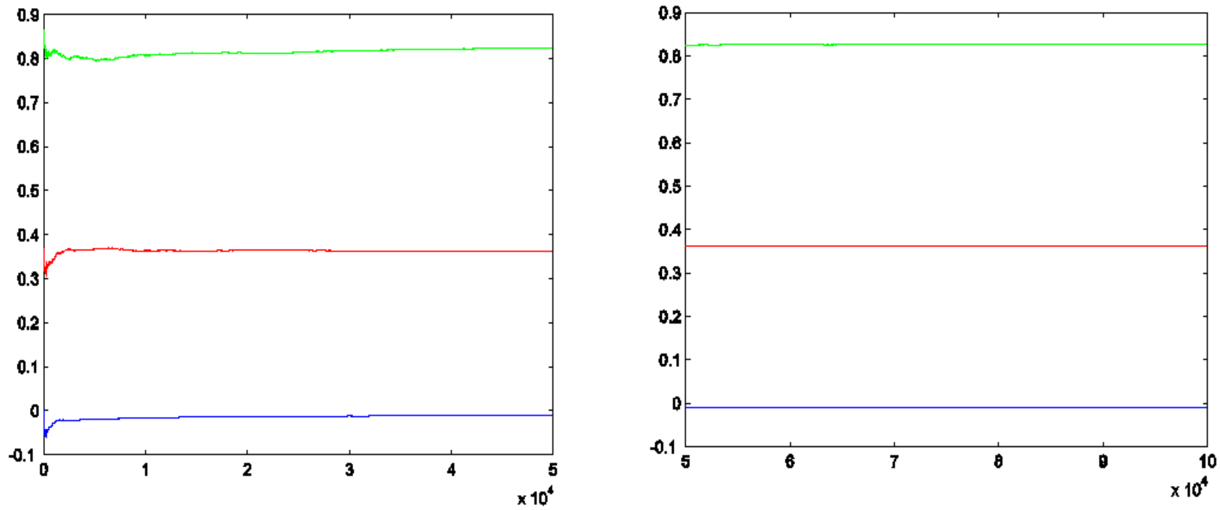

Figure 2: Evolution of three parameter estimates for the consumption equation from a single run of the RLS algorithm using the linearized model, linear scale. Left panel periods 1-50,000 periods, right panel periods 50,000-100,000.

in which the precise MSV RE solution has only been approximately learned.

For the non-linear, real-time PE algorithm, to assess convergence we have to first estimate θ^*. We do this using the conventional, off-line PE algorithm where there is no role for exogenous shocks or learning dynamics in the determination of the six parameter values of θ^*. Using a sample size of 200,000 observations, we find that

$$\theta^{*\prime} = [0.7530; -0.7363; -1.6712; 0.0278; 0.0478; -0.0728],$$

which we use to assess convergence of the real-time, on-line PE algorithm, which is affected by exogenous shocks and learning dynamics.

The evolution of the 6 parameter estimates of the nonlinear expectation approximating function for the consumption equation over the 100,000 iterations of a representative run of our real-time PE algorithm is shown in Figure 3. Again convergence obtains according to our convergence criterion after a few hundred periods, but we show the evolution of the coefficient estimates over the 100,000 periods allowed. Following 100,000 iterations, the

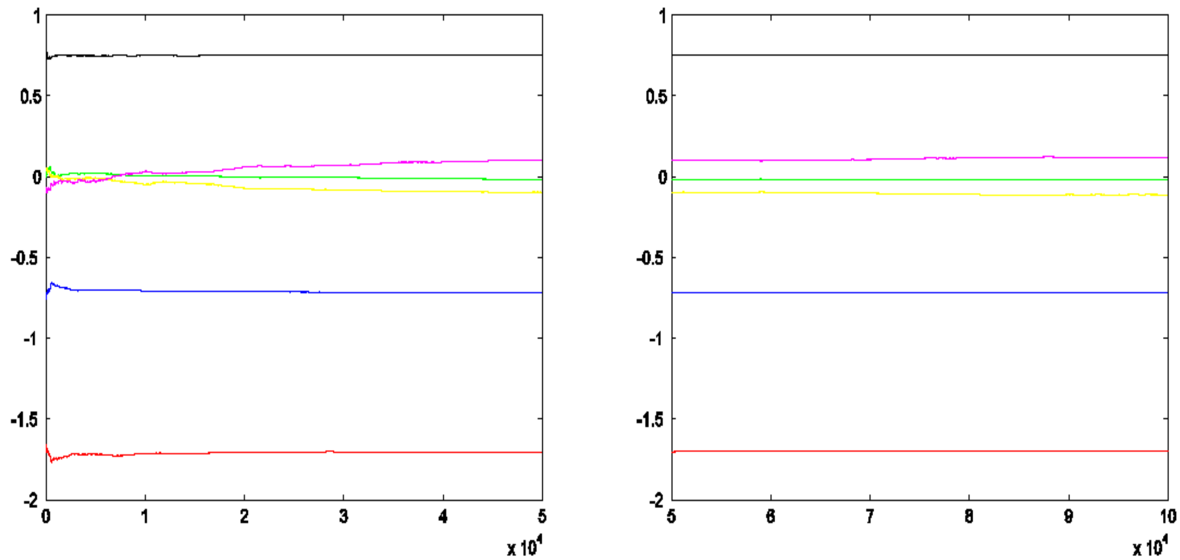

Figure 3: Evolution of six parameter estimates from a single run of the real-time PE algorithm using the nonlinear model, linear scale. Left panel periods 1-50,000, Right panel periods 50,000-100,000.

estimated parameters from the real-time PE run depicted in Figure 3 are given by:

$$\theta'_{t=100,000} = [0.7846; -0.7025; -1.7488; -0.0158; -0.0430; 0.0662],$$

which is a reasonably good approximation to θ^*; recall that in this case θ^* is itself an approximation to the true RE solution and that the best our real-time PE algorithm can do (using an exponential polynomial approximation) is to get arbitrarily close to the RE solution.

In response to the second question we posed, Table 1 reports for each of the 9 model parameterizations that we considered, the mean and variance in the number of periods (from 100 runs) required before convergence was declared using our convergence criterion, $\max_i\{|\theta_t^i - \theta_{t-1}^i|\} < \tau$, where i indexes the coefficients to be learned. In comparing the Linearized RLS model with the Nonlinear Real Time-PE approach, we consider three different versions of the linearized RLS model. The first version, denoted 'Linearized RLS, With Constants' in Table 1 is the most standard version found in the literature where there

Model Parameterization[a]	Linearized RLS						Nonlinear Real Time-PE	
	With Constants		No Constants		With $\hat{k}$, $\hat{z}$ Known			
	Avg. Time	Var.	Avg. Time	Var.	Avg. Time	Var.	Avg. Time	Var.
Baseline[b]	653.53	8.04×10^4	335.75	2.08×10^4	266.43	2.18×10^4	433.24	8.79×10^4
$\delta = \sigma = 1$	1274.70	3.81×10^5	574.60	2.30×10^5	267.21	2.35×10^4	65.75	3.20×10^3
$\delta = 0.2$	534.61	5.20×10^4	288.85	1.95×10^4	258.24	2.31×10^4	392.32	6.39×10^4
$\delta = 0.1$	391.18	2.50×10^4	217.54	1.24×10^4	238.62	1.57×10^4	370.17	7.16×10^4
$\delta = 0.025$	827.07	2.22×10^6	642.20	1.33×10^6	657.52	1.45×10^6	309.30	6.43×10^4
$\alpha = 0.5$	499.76	4.19×10^4	286.13	2.05×10^4	242.29	2.29×10^4	623.16	6.68×10^5
$\sigma = 1.0$	676.27	7.02×10^4	308.08	2.20×10^4	233.40	1.53×10^4	230.57	1.84×10^4
$\sigma = 0.5$	580.24	6.03×10^4	295.74	2.32×10^4	205.01	1.45×10^4	69.61	2.21×10^3
$\sigma_\varepsilon^2 = .02$	410.83	5.07×10^4	220.45	1.62×10^4	194.76	1.51×10^4	258.85	2.27×10^4

a. Parameterizations other than the baseline case differ from the baseline case only in the parameter choice(s) indicated.
b. Baseline parameterization: $\delta = .3$, $\alpha = .33$, $\rho = .95$, $\beta = .98$, $\sigma = 2$, $\sigma_\varepsilon^2 = .1$.

Table 1: Average Time to Convergence and Variance from 100 Simulations of the Linearized-RLS with/without constant terms or with knowledge of $\hat{k}$ and $\hat{z}$ and the Nonliner Real Time-PE Models Under Various Parameterizations

is a constant term (vector) in the PLM as in (9). In a second version of the RLS model denoted 'Linearized RLS, No Constant' in Table 1, we eliminate the constant terms, so the number of parameters that need to be estimated in the linear case is now 5. The latter case may be a more appropriate basis for comparison as the linearized model is stated in terms of deviations from steady state values so agents should eventually learn that the value of the constant terms in their PLM are all zero; the need to learn this fact may slow the RLS algorithm down relative to the case where the constant term is eliminated from the learning specification. Further, in this case the RLS algorithm requires estimation of just 5 parameters which is one less than the real-time PE model, which always involves 6 parameters in our application and specification. Finally, we consider a third version of the RLS model with a constant term but where we endow agents with knowledge of the laws of motion for capital, $\hat{k}$ and the shock process $\hat{z}$ so that they are only learning the 3 parameters of the linearized consumption equation (of (9)). Using our Real Time-PE approach, agents are also only learning the parameters of the consumption equation. Thus again, our aim here is to attempt to put the two methods on a more equal setting for comparison.

Based on the numbers reported in Table 1 we have the following findings:

1. The linearized RLS algorithm without the constant term or where the law of motion for capital and the shock process are known always converges faster than the linearized RLS algorithm with the constant term and where the process for capital and the shock terms must be learned, which is not so surprising as it can take time to learn the additional coefficients in the latter specification of RLS learning.

2. The average time to convergence using the real-time, nonlinear PE algorithm is faster than using the linearized RLS algorithm without the constant term, or without the need to learn the capital stock or shock process in 4 out of 9 model parameterizations. Relative to the RLS algorithm where agents learn the constant and processes for capital and the shock term, the real-time nonlinear PE algorithm is faster in 8 of the 9 model parameterizations.

3. A low (but empirically more plausible) value for δ, (0.025) greatly increases the time to convergence of all three versions of the RLS algorithm relative to convergence time under the baseline parameterization, but the same is not true of the real–time PE algorithm.

3. A higher α (0.5) reduces the time to convergence for all three versions of the RLS algorithm relative to the baseline parameterization, but not for the real–time PE algorithm.

4. A much lower value for σ (0.5) reduces the time to convergence for both RLS and Real Time-PE algorithms relative to the baseline parameterization $\sigma = 2$, but this reduction is significantly greater for the real-time PE algorithm.

5. The time to convergence for both the algorithms depends on σ_ε^2. A lower $\sigma_\varepsilon^2 = 0.02$ decreases the time to convergence, relative to the baseline case where $\sigma_\varepsilon^2 = 0.10$.

6. For the special case of $\delta = \sigma = 1$, where a closed-form nonlinear solution is available, the real-time PE algorithm delivers its best convergence performance relative to the other parameterizations considered in Table 1, converging on average in just 65.75 periods. By contrast, this same parameterization results in one of the poorest performances in terms of speed of convergence for all three versions of the RLS algorithm, relative to the other parameterizations considered in Table 1.

Baseline Parameterization	Linearized RLS with Constants	Linearized RLS with k, z known	Nonlinear Real Time-PE
Average time	641.39	266.43	433.24
Median time	581.5	243	396.5
Max time	1451	722	6492
Min time	134	8	27
25th percentile	442.5	159	216.5
75th percentile	834.5	356	668.0

Table 2: Information on the distribution of convergence times under the two real–time algorithms for the baseline parameterization only.

7. For the baseline parameterization, the real–time PE algorithm is faster to converge than the RLS algorithm with a constant, but slower than the RLS algorithm without a constant or when there is no need to learn the law of motion for the capital stock and shock process. Some information on the distribution of convergence times under the two algorithms for the baseline parameterization is provided in Table 2.

5.2 Robustness and Implementation Issues

As a further robustness check we consider the mean time to convergence of the two real-time learning algorithms under various combinations for the values of σ and δ, holding all other parameters fixed at their baseline values. The results of this exercise are reported in Tables 3-4.

	δ			
σ	.025	.1	.2	.3
0.5	129.55 (7.68×10^3)	58.82 (2.44×10^3)	61.84 (2.27×10^3)	69.61 (2.21×10^3)
1.0	243.10 (3.38×10^4)	223.83 (2.29×10^4)	226.55 (2.11×10^4)	230.57 (1.84×10^4)
1.5	308.93 (4.97×10^4)	332.29 (4.85×10^4)	326.54 (4.18×10^4)	333.04 (4.03×10^4)
2.0	309.30 (6.43×10^4)	370.17 (7.16×10^4)	392.32 (6.39×10^4)	433.24 (8.79×10^4)

Table 3: Nonlinear PE algorithm: Mean Time to convergence for various σ-δ combinations (Variance in brackets)

	δ			
σ	.025	.1	.2	.3
0.5	225.13 (1.62×10^4)	288.65 (1.42×10^4)	411.19 (2.61×10^4)	580.24 (6.03×10^4)
1.0	363.17 (2.77×10^4)	358.41 (2.14×10^4)	482.70 (4.71×10^4)	676.27 (7.02×10^4)
1.5	561.08 (6.49×10^5)	363.56 (2.56×10^4)	494.48 (3.70×10^4)	632.03 (7.96×10^4)
2.0	827.07 (2.22×10^6)	391.18 (2.50×10^4)	534.61 (5.20×10^4)	653.53 (8.04×10^4)

Table 4: Linearized RLS algorithm: Mean Time to convergence for various σ-δ combinations (Variance in brackets)

We observe two general phenomena from this exercise. First, for any given value of δ, an increase in σ, implying greater risk aversion on the part of the representative agent household (and increasing relative nonlinearity of the system) always results in an increase in the mean time to convergence under both real-time algorithms. Second, holding σ fixed, an increase in δ generally, though not always monotonically leads to an increase in the mean time to convergence. Comparing Tables 3 and 4 we see that the Real Time-PE approach generally outperforms the linear RLS algorithm in mean time to convergence. Taken together, these findings indicate that, for the optimal growth model, the nonlinear, parameterized expectations approach can be implemented in real time without much reduction in the speed of convergence, and in several instances, the real-time PE algorithm is considerably faster. The findings confirm 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 the exact solution in a linearized setting.

We wish to further address two implementation issues for our nonlinear Real Time-PE algorithm which concern the choice of the initial parameter vector and the approximating polynomial function. We discuss each in turn.

The initial parameter vector in all of our simulations of the Real Time-PE algorithm was chosen to be 10 percent off, i.e., 10% higher than the RE solution parameter vector θ^*. There is in principle no reason that we cannot choose different initial values and assess the extent to which convergence obtains under such conditions. Indeed, since the Real Time-

PE learning system has not been linearized, variations in the initial conditions can serve as check on the "global" stability of the rational expectations solution under our real-time PE learning dynamic; by contrast, the linearized system is only valid within a neighborhood of the steady state solution and so initial parameter values cannot be very far off from the RE solution values.

Table 5 below indicates the frequency with which the Real Time-PE algorithm converges to the RE vector θ^* for various different initial starting values for θ_0, expressed as percent deviations, + or − from θ^*.

% Deviation of initial conditions from θ^*	Frequency of Non-convergent runs
+10	21 %
+20	24 %
+40	31 %
+60	33 %
+80	33 %
−20	26 %
−40	32 %
−60	89 %
−80	96 %
All zero	98 %

Table 5: Frequency of non-convergent runs out of 100 for the Real Time-PE algorithm for different initial parameter vector conditions

Non-convergence obtains when the capital stock becomes negative which subsequently generates a sequence of imaginary numbers. When such an event occurred, we declared that non-convergence of the algorithm had occurred and we ended that run. The results in Table 5 reveal an absence of global stability of the Real Time-PE algorithm as the frequencies of non-convergence are always non-zero. However, we note a couple of interesting regularities. First, positive deviations of initial parameterizations from the RE parameter vector are less likely to lead to non-convergence than are negative deviations. Second, it is possible to use initializations that are quite far away, e.g., $\pm 40\%$ of θ^* without much reduction in the frequency of convergence relative to the baseline initialization of a +10 percent deviation. Finally, initialization of the system using a zero vector for θ_0 ("All zero" in Table 5) is, perhaps not surprisingly, the worst initialization scheme among those we considered.

The choice of the polynomial function used to approximate expectations is also an important implementation issue. In our simulations, we chose an exponential polynomial of order 2 as this is the minimal polynomial order that allows for nonlinearities that are present in the system. However, even this specification might be regarded as over-parameterized relative to the RE solution; recall that the last three coefficients in $\theta^{*\prime}$ are close to zero. More generally, learning agents might not know the appropriate choice (order) of the nonlinear approximating function so this specification choice could also be part of their learning process.[11]

As a robustness check, we considered the case where agents used a third-order polynomial approximation. Specifically, we studied the case where the exponential polynomial function

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

has

$$\begin{aligned}
\theta' &= [\theta_0\ \theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6\ \theta_7\ \theta_8\ \theta_9\ \theta_{10}], \\
x_t' &= [1\ \ \log(k_t)\ \ \log(z_t) \\
&\quad \log(k_t)^2\ \ \log(z_t)^2\ \ \log(k_t)\log(z_t) \\
&\quad \log(k_t)^3\ \ \log(z_t)^3\ \ \log(k_t)^2\log(z_t)\ \ \log(k_t)\log(z_t)^2].
\end{aligned}$$

This case includes many more terms, i.e., $\log(k_t)^3$, $\log(z_t)^3$, $\log(k_t)^2\log(z_t)$, and $\log(k_t)\log(z_t)^2$ than the second-order polynomial approximation, and so this specification can be regarded as even more over-parameterized relative to the RE solution.[12] With this specification for

[11] Analogously, in the linear RLS learning literature, there is an issue of whether perceived laws of motion should be exactly specified or whether over-parameterized models should be considered.

[12] Another possible miss-specification to consider would be to use an underparameterized polynomial approximation, with fewer terms than necessary to approximate the nonlinear expectation function. Such specifications would not converge to RE solutions but might converge to self-confirming equilibria. As we are interested in the learning of RE solutions, we do not consider such underparameterized models here, though this would be a good topic for future research.

expectations and using the baseline parameterization of the model, we first found

$$\theta^{*\prime} = [0.7534, -0.7299, -1.6769, 0.0269, 0.0477, -0.0710, -0.0025, 0.0069, 0.0092, -0.0130]$$

using Collard's algorithm and a sample size of 200,000 observations. Notice that the additional terms added by the higher order polynomial approximation are estimated to be close to zero and so this third-order polynomial approximation does not differ much relative to the second-order polynomial. Simulations of the real-time PE algorithm with the third-order polynomial specification resulted in non-convergence with a very high frequency, approximately 80%. However, in those instances where convergence obtained, convergence was actually faster relative to the second-order polynomial case, and the extra parameters of the third-order polynomial approximation were essentially equal to zero.

If non-convergence (as evidenced by the appearance of imaginary numbers) is indicative of model miss-specification, then learning agents would presumably revisit either the choice of their approximating function or their initialization of the associated parameter vector or both. For this reason, we think it is reasonable to focus on convergent outcomes (as we have done in our main findings section) which are more frequent with low-order polynomial approximations and initial parameter vectors that are not too far off from RE solution values.

6 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 an important micro-foundation for the rational expectations hypothesis. However, much of this literature presumes that agents use a linear specification when forming expectations of future endogenous variables, even though the system in which agent are operating is more typically nonlinear. The appeal of linear forecasting rules is that in linearized systems it is possible to find analytic conditions under which adaptive processes involving these linear forecast rules converge to RE solutions of the linearized system.

Our approach is to relax the assumption that agents use a linear, perceived law of motion and instead suppose that agents form forecasts using a nonlinear, parameterized expectations approach. Furthermore, we incorporate these nonlinear forecasts into the nonlinear system, i.e., we do not consider the linearized system. While the parameterized expectations method was originally devised as an off-line means of finding 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 more global rival to other real-time learning algorithms used to assess local stability 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 converges to (a neighborhood of) the rational expectations equilibrium of the optimal growth model that we study under a variety of different parameterizations of that model. Mean convergence times and the variance in those times are comparable between the real-time PE approach and the more conventional, real-time RLS approach using a linearized version of the model and a linear specification for expectation formation; sometimes the PE algorithm is slower and sometimes it is faster that the RLS learning algorithm using the linearized system, depending on model parameters. Our simulation exercises consider other important issues regarding the choice of the approximating nonlinear polynomial function and the initial conditions for the parameterization of that function.

Our results suggest that nonlinear real-time learning models are a plausible alternative to linear learning models particularly in environments where the system that agents are seeking to learn is also nonlinear. We note that the application we have considered is a nonlinear system with a unique, E-stable rational expectations equilibrium. It is still unknown how well our real-time PE approach would work in non-linear systems where the equilibrium is known to be E-unstable under adaptive (least squares) learning or where there are multiple rational expectations equilibria. We leave that exercise to future research.

Appendix

Linearization of the growth model

The system to be linearized consists of equations (3)-(6). In linearizing this system we make use of the transformation, $\hat{x}_t = \ln\left(\frac{x_t}{\bar{x}}\right)$, where $\bar{x}$ is the steady state value of variable x, i.e., $\hat{x}_t$ denotes the deviation of x_t from its steady state value, $\bar{x}$. The linearization approach we adopt follows Uhlig (1999). In particular, we can combine linearized versions of equations (3) and (5) to obtain:

$$\hat{k}_{t+1} = \gamma_z \hat{z}_t + \gamma_k \hat{k}_t + \gamma_c \hat{c}_t, \tag{14}$$

where $\gamma_z = \frac{1-\beta(1-\delta)}{\alpha\beta}$, $\gamma_k = \frac{1}{\beta}$ and $\gamma_c = \frac{-1+\beta-\beta\delta(1-\alpha)}{\alpha\beta}$. Similarly, we can combine linearized versions of equations (4) and (5) to obtain:

$$\hat{c}_t = \phi_z E_t \hat{z}_{t+1} + \phi_k E_t \hat{k}_{t+1} + \phi_c E_t \hat{c}_{t+1}, \tag{15}$$

where $\phi_z = (\beta(1-\delta)-1)/\sigma$, $\phi_k = (\alpha-1)(\beta(1-\delta)-1)/\sigma$ and $\phi_c = 1$. Finally, linearizing equation (6) yields:

$$\hat{z}_t = \rho \hat{z}_{t-1} + \hat{\epsilon}_t. \tag{16}$$

References

[1] Bullard, J., Evans, G.W., Honkapohja, S. (2010). A model of near-rational exuberance. *Macroeconomic Dynamics* 14, 166-188.

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

[3] Collard, F. (2002). Unpublished lecture notes number 8: Parameterized expectations algorithm and associated Matlab code available at: http://fabcol.free.fr/.

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

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

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

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

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

[9] Evans, G.W., Honkapohja S., and Marimon, R. (2007) "Stable Sunspot Equilibria in a Cash-in-Advance Economy," *The B.E. Journal of Macroeconomics (Advances)* 7:1, Article 3.

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

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

[12] Hommes, C., Sorger, G. (1998). Consistent expectations equilibria. *Macroeconomic Dynamics* 2, 287-321.

[13] Lansing, K.J. (2006). Lock-in of extrapolative expectations in an asset pricing model. *Macroeconomic Dynamics* 10, 317–348.

[14] Ljung, L. (1975). Theorems for the asymptotic analysis of recursive stochastic algorithms. Report 7522, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

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

[16] 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.

[17] 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.

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

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

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

[21] 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.

[22] 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.