Journal of Business & Economic Statistics, January 1990, Vol. 8, No. 1

Solving the Stochastic Growth Model by Parameterizing Expectations

Wouter J. den Haan and Albert Marcet
Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213

This article describes a method for solving the one-good stochastic growth model by parameterizing the expectations part of the stochastic Euler equation. The conditional expectation is specified as a function of the state of the system, and the parameters of that function are estimated to solve the model. The article includes a discussion of how to find the parameters of the function and determine systematically the complexity of the functional form necessary to solve the model.

KEY WORDS: Dynamic model; Fixed point; Numerical solution; Polynomial approximation.

1. INTRODUCTION

In this note, we will describe the calculations that we performed to simulate the following growth model:

$$\max E_0 \sum_{t=0}^{\infty} \beta^t [c_t^{1-\tau}]/(1-\tau),$$

subject to

$$c_t + k_t = \theta_t k_{t-1}^{\alpha} + \mu k_{t-1}. \tag{1}$$

We have used the method of parameterizing expectations. This method is described in detail in Marcet (1988), and it has been applied successfully in several models. Marshall (1988) solved a model with transaction costs and growth. Den Haan (1988) calculated optimal monetary-policy rules in a model with a shopping-time technology. Ketterer and Marcet (1988) solved asset-pricing models with heterogeneous agents that trade different types of securities (namely, bonds, stocks, options, and futures); unlike representative-agent models, these papers generate actual trading of securities. In Den Haan and Marcet (1989), we proposed a method for testing for accuracy of simulations; the parameterized expectations procedure does very well in these tests; in that paper we also solved the time-to-build model of Kydland and Prescott (1982) and reproduced their results.

The equations that describe the equilibrium in the preceding growth model are the production constraint (1) and

$$c_t^{-\tau} = \beta E_t[c_{t+1}^{-\tau}(\theta_{t+1}\alpha k_t^{\alpha-1} + \mu)] \tag{2}$$

and

$$\log \theta_t = \rho \log \theta_{t-1} + \varepsilon_t. \tag{3}$$

Here (2) is the first-order condition when we differentiate with respect to the capital stock and (3) is simply the law of motion for the productivity shock. Note that we have introduced a depreciation parameter μ (the original model sets $\mu = 1$).

We use the observation that the conditional expectation in the right side of (2) is a function $g : R_+^2 \to R_+$ of the state variables (k_{t-1}, θ_t), as a basis for substituting that conditional expectation with a function $\psi(k_{t-1}, \theta_t; \delta_f)$, where ψ (the functional form) and δ_f (the vector of parameters) will be chosen to make $\psi(\cdot; \delta_f)$ as close as possible to g. The crucial part of this solution procedure is finding such ψ and δ_f.

2. CHOICE OF ψ

By letting ψ belong to a class of functions that can approximate any function arbitrarily well, we can, in principle, approximate g. For example, ψ could be a polynomial; increasing the order of the polynomial progressively, we can obtain arbitrary accuracy. In addition, observing how the equilibrium changes when we increase the order of the polynomial, we can have some idea of how close we are to g.

Our first choice for ψ is a power function, $\psi(k_{t-1}, \theta_t; \delta) = \delta_1 k_{t-1}^{\delta_2} \theta_t^{\delta_3}$. One reason for this choice is that its image is positive, as is the image of the function g that we want to approximate. Strictly speaking, the image of ψ is positive iff $\delta_1 > 0$. Given the procedure that we use to find δ_f (see next paragraph), however, δ_1 is always positive in the iterations that lead to calculating δ_f as long as the initial δ_1, $\delta_1^0 > 0$.

Moreover, letting $P_n(x)$ denote a polynomial of degree n on the vector x, the preceding power function can be rewritten as $\exp(P_1(\log k_{t-1}, \log \theta_t))$. Since it can be shown that functions of the form $\exp(P_n(x))$ can approximate any function mapping R_+^2 into R_+, and g is such a function, by letting $n \to \infty$, we can approximate the equilibrium arbitrarily well.

3. CHOICE OF δ_f

Let $\{c_t(\delta), k_t(\delta)\}$ be the sequence of consumption and capital that solves

$$c_t^{-\tau} = \beta\psi(k_{t-1}, \theta_t; \delta) \tag{4}$$

and Equation (1) for all t, for a given δ, and for a given

realization of θ drawn from (3). Equation (4) is the Euler equation (2) with ψ in place of the conditional expectation. For a given ψ and δ, solving for k and c is very easy, since we have two equations and two unknowns at each period; first, we find c_t from (4) and then we find k_t from (1).

Define $S : R^m \to R^m$, where m is the dimension of δ, as

$$S(\delta) = \underset{\bar{\delta}}{\operatorname{argmin}} E[c_{t+1}^{-\tau}(\delta)(\theta_{t+1}\alpha k_t^{\alpha-1}(\delta) + \mu) - \psi(k_{t-1}(\delta), \theta_t; \bar{\delta})].$$

We choose the parameter δ_f to satisfy $\delta_f = S(\delta_f)$. This guarantees that if agents use $\psi(k_{t-1}, \theta_t; \delta_f)$ as their expectation function, then δ_f *is* the best parameter they could use in the sense that it minimizes the mean squared error. The choice of mean squared error as the measure for good prediction is justified because the conditional expectation satisfies this criterion.

4. CALCULATING δ_f

The fixed point δ_f is calculated with the following iterative procedure. First, generate a series for θ_t of length T (T has to be large enough that the parameters in the regressions that we will perform in the next step are calculated with accuracy). Note that this series is drawn only once. Then we choose an initial $\delta^0 \in R^3$ and calculate $\{c_t(\delta^0), k_t(\delta^0)\}_{t=1}^T$. Next, we run a nonlinear least squares regression (NLR) of $c_{t+1}^{-\tau}(\theta_{t+1}\alpha k_t^{\alpha-1} + \mu)$ on the power function ψ. This is our approximation to $S(\delta^0)$; it is easy to show that the result of the NLR converges to $S(\delta^0)$ as T goes to infinity.

Now $\delta^1, \delta^2, \ldots$ are given by the following iterative scheme:

$$\delta^v = (1 - \lambda)\delta^{v-1} + \lambda S(\delta^{v-1}), \qquad v = 1, 2, \ldots \quad (5)$$

for a $\lambda \in (0, 1]$ appropriately chosen. Thus with δ^1 we solve (1) and (4) again, run an NLR to find $S(\delta^1)$, and iterate on (5). We stop the iterations when δ^v is close to $S(\delta^v)$.

To run the NLR, we use the procedure of running a linear regression on a first-order Taylor expansion of ψ. Since we have good initial conditions for these NLR parameters [namely, $S(\delta)$ from the previous iteration], calculating each $S(\delta)$ involves running only a few ordinary least squares (OLS) regressions (from two to four). A more detailed description is given in the Appendix.

The parameter λ governs the adjustment speed of the algorithm. With low λ, the algorithm converges for almost all of the models that we have tried, but a low λ makes adjustments in δ very small and convergence is very slow. After experimenting with each model, the researcher has some idea of what is the highest possible λ that can be used.

For the present model, $\{\delta^v\}$ converges even with $\lambda = 1$, so this is the value we used. In more complicated models, λ usually has to be smaller.

The iterations will converge for λ small enough if a certain least squares learning mechanism converges to the rational-expectations equilibrium; that is, for those models in which the rational-expectations equilibrium is stable under learning, the preceding iterative scheme converges. We know from the learning literature that in some cases the least squares learning mechanism is unstable, and in these models the algorithm defined by (5) will not converge. Nevertheless, these models are more the exception than the rule; in fact, (5) has converged in all of the models that we have simulated with this procedure. Finally, it may be desirable to concentrate our attention on those models where the rational-expectations equilibrium can be learned, and this algorithm does that automatically. These points are discussed in detail in Marcet (1988).

The algorithm described by (5) is very easy to update; the iterations do not involve finding the best direction, which is usually very costly. Equation (5) has only good local convergence properties, however, and it is necessary to use good initial conditions for δ^0. To do this in a systematic way, we start at the solution to the model when $\tau = 1$, $\mu = 0$. This is the model of Brock and Mirman (1972), and we know δ_f analytically. Then we let μ go from 0 to 1 in 10 steps, calculate δ_f for each step, and use it as initial condition for the next μ. The δ_f's we obtain for a first-order power function are given in Table 1.

The values of the remaining parameters in this homotopy are $\beta = .95$ and $\sigma_\varepsilon = .1$; $\lambda = .5$ for $u \leq .5$, and $\lambda = 1.0$ for $u > .5$.

Twenty-five hundred observations for calculating the NLR and four digits of accuracy in the fixed point were used. The calculations were made with a Compaq deskpro 386, 25 megahertz, using a Weitek mathematical coprocessor. We used a FORTRAN program and double precision in all real variables. The FORTRAN compiler is NDP-386, version 1.4, by Microway, Inc., running on MS-DOS, version 3, in extended mode. When the program was run on the same machine using a Ryan McFarland FORTRAN compiler (which is a 286-based compiler), the computing times tripled.

It is worth pointing out that, given these δ_f's, readers can obtain simulations on their own computers very

Table 1. Fixed-Point Parameters for Different Depreciation Rates and Computing Times

μ	δ_{1f}	δ_{2f}	δ_{3f}	Computing time
.0	1.53	−.33	−1.00	
.1	1.49	−.35	−.97	10 seconds
.2	1.45	−.37	−.94	8 seconds
.3	1.42	−.39	−.91	10 seconds
.4	1.40	−.41	−.87	8 seconds
.5	1.39	−.43	−.83	12 seconds
.6	1.40	−.46	−.78	8 seconds
.7	1.44	−.49	−.72	8 seconds
.8	1.53	−.52	−.65	18 seconds
.9	1.74	−.56	−.55	25 seconds
1.0	2.47	−.65	−.36	65 seconds

easily. Finding new δ_f's for different parameter values is a little harder, but readers will find the reported δ_f's useful as initial conditions.

An algorithm from stochastic approximation is proposed in Marcet (1988). This algorithm is very fast although it is less stable than the preceding algorithm. Once researchers feel comfortable with the solution of the model, they can use the stochastic algorithm to perform more computer-intensive tasks like estimation by simulation.

5. ACCURACY

One way of checking for accuracy in the solutions is to increase the degree of the exponentiated polynomial and use

$$\begin{aligned}\psi(k_{t-1}, \theta_t; \delta) &= \exp(P_2(\log k_{t-1}, \log \theta_t)) \\ &= \exp(\delta_1 + \delta_2 \log k_{t-1} + \delta_3 \log \theta_t + \delta_4(\log \theta_t)^2 \\ &\quad + \delta_5(\log k_{t-1})(\log \theta_t) + \delta_6(\log k_{t-1})^2) \qquad (6)\end{aligned}$$

in place of the first-order power function.

It turns out that the terms $(\log k_{t-1})(\log \theta_t)$ and $(\log k_{t-1})^2$ are almost perfectly collinear with the others. More precisely, the matrix

$$E\left(\frac{\partial \psi_t}{\partial \delta}\frac{\partial \psi_t}{\partial \delta'}\right)$$

is close to singular due to the derivatives with respect to δ_5 and δ_6.

This is, in fact, a fortunate situation. It just means that these terms are redundant, and they can be dropped from ψ_t without losing any predictive power. Hence for a second-degree polynomial we only need to solve for δ_{if} $(i = 1, \ldots, 4)$, and we can set $\delta_{5f} = \delta_{6f} = 0$.

To get good initial conditions for δ_f, we take the solution generated with the first-order polynomial, run a regression with the term $(\log \theta_t)^2$ included, and use that as our initial condition.

This can be done with higher-order polynomials in ψ. Next we report the (absolute) differences in the solution between (c_t, k_t) from a first- and second-degree polynomial and the (absolute) difference between a second and third degree. The latter are very small; for example, in case 5 the average difference is .01% in going from second to third order. A difference of .01% corresponds with four digits of accuracy in the series for consumption and capital. Such small differences can be taken as an indication that the solution with a second-order polynomial is a good approximation. The results in Table 2 were calculated using 1,500 observations.

The results submitted were generated with a second-degree polynomial of the type discussed in this section.

Den Haan and Marcet (1989) proposed a test for accuracy in simulations and evaluated the accuracy of solutions obtained by parameterizing expectations; they solved several growth models and showed some examples in which higher-order polynomials improve the quality of the approximation substantially.

Table 2. Differences With Higher-Order Polynomials

	From first to second order		From second to third order	
	k	c	k	c
Case 1 (high variance of ε)				
Maximum	2.62%	2.27%	.63%	.86%
Average	.38%	.29%	.07%	.05%
Case 5 (low variance of ε)				
Maximum	.17%	.12%	.11%	.17%
Average	.04%	.02%	.01%	.01%

ACKNOWLEDGMENTS

Den Haan acknowledges fellowship support from the Earhart Foundation and the John M. Olin Foundation. Marcet's research was supported by a grant from the National Science Foundation.

APPENDIX: THE PROCEDURE USED FOR THE NONLINEAR REGRESSIONS INVOLVED IN CALCULATING $S(\delta)$

Given δ, and series $\{c_t(\delta), k_t(\delta), \theta_t\}_{t=1}^T$, the problem is to find a parameter $\gamma^{\min}$ that minimizes

$$(1/T)\sum_{t=1}^{T}[\phi_t - \psi(k_{t-1}(\delta), \theta_t; \gamma)]^2,$$

where $\phi_t = c_{t+1}^{-\tau_1}(\delta)(\alpha\theta_{t+1}k_t^{\alpha-1}(\delta) + \mu)$.

We know that, for large enough T, $\gamma^{\min}$ is a good approximation to $S(\delta)$. We obtain series $\{\gamma^n\}$ with the following iterative scheme: Let

$$\frac{\partial \psi_t(\bar{\gamma})}{\partial \gamma} = \left.\frac{\partial \psi(k_{t-1}(\delta), \theta_t; \gamma)}{\partial \gamma}\right|_{\gamma = \bar{\gamma}}.$$

Given a γ^n, we run a *linear* regression in which the dependent variable is given by

$$\left[\phi_t - \psi(k_{t-1}(\delta), \theta_t; \gamma^n) + (\gamma^n)'\frac{\partial \psi_t(\gamma^n)}{\partial \gamma}\right],$$

and the independent variables are $\partial\psi_t(\gamma)/\partial\gamma$. Note that we have taken a first-order approximation of ψ_t around γ^n and rearranged terms. Then γ^{n+1} is given by the resulting coefficients in this OLS regression. It turns out that γ^n converges to $\gamma^{\min}$ as n goes to infinity. [This procedure was discussed in more detail by Pindyck and Rubinfeld (1981, sec. 9.4.1).] This is a version of a Gauss–Newton algorithm, so if γ^n is close to the limit, convergence is reached in one iteration. In our case, only two to four *linear* regressions were needed for each NLR.

To speed up calculations, it is best to use as the initial condition for these iterations the result of the previous NLR in Algorithm (5). More precisely, if we are trying to calculate $S(\delta^v)$, we could set $\gamma^0 = S(\delta^{v-1})$.

Moreover, the derivative of ψ with respect to γ can

be computed quite efficiently. For example, in the case that we parameterize ψ as $\psi(k_{t-1}, \theta_t; \delta) = \delta_1 k_{t-1}^{\delta_2} \theta_t^{\delta_3}$, we have

$$\frac{\partial \psi_t(\gamma)}{\partial \gamma} = \psi(k_{t-1}, \theta_t; \gamma) \begin{bmatrix} 1/\gamma_1 \\ \log(k_{t-1}) \\ \log(\theta_t) \end{bmatrix},$$

and for a different γ only the first element of the vector in the right side and the value of ψ change. Hence, if we store the logarithms of k and θ before we run the NLR, we can calculate this derivative very quickly.

[*Received December 1988.*]

REFERENCES

Brock, W. A., and Mirman, L. J. (1972), "Optimal Economic Growth and Uncertainty: The Discounted Case," *Journal of Economic Theory*, 4, 479–513.

Den Haan, W. J. (1988), "The Optimal Inflation Path in a Sidrauski-Type Model With Uncertainty," unpublished manuscript, Carnegie Mellon University, Graduate School of Industrial Administration.

Den Haan, W. J., and Marcet, A. (1989), "Accuracy in Simulations," unpublished manuscript, Carnegie Mellon University, Graduate School of Industrial Administration.

Ketterer, J., and Marcet, A. (1988), "Introducing Derivative Securities in an Equilibrium Model," unpublished manuscript, Carnegie Mellon University, Graduate School of Industrial Administration.

Kydland, F. E., and Prescott, E. C. (1982), "Time to Build and Aggregate Fluctuations," *Econometrica*, 50, 1345–1370.

Marcet, A. (1988), "Solving Non-linear Models by Parameterizing Expectations," unpublished manuscript, Carnegie Mellon University, Graduate School of Industrial Administration.

Marshall, D. A. (1988), "Inflation and Asset Returns in a Monetary Economy," unpublished Ph.D. dissertation, Carnegie Mellon University, Graduate School of Industrial Administration.

Pindyck, R. S., and Rubinfeld, D. L. (1981), *Econometric Models and Economic Forecasts* (2nd ed.), New York: McGraw-Hill.