Macroeconomic Dynamics, **1**, 1997, 45–75. Printed in the United States of America.

SOLVING LARGE-SCALE RATIONAL-EXPECTATIONS MODELS

JESS GASPAR
University of Chicago

KENNETH L. JUDD
Hoover Institution

We explore alternative approaches to numerical solutions of large rational-expectations models. We discuss and compare several current alternatives, focusing on the trade-offs in accuracy, space, and speed. The models range from representative-agent models with many goods and capital stocks, to models of heterogeneous agents with complete or incomplete asset markets. The methods include perturbation and projection methods. We show that these methods are capable of analyzing moderately large models even when we use only elementary, general-purpose numerical methods.

Keywords: Rational-Expectations Models, Representative Agents, Heterogeneous Agents, Computational Methods, Perturbation Methods, Projection Methods

1. INTRODUCTION

The study of macroeconomic dynamics has become substantially more sophisticated over the past 20 years. The first models were simple models with one commodity, serving as both the consumption and the investment good, and with one type of agent as in the representative-agent models, or very similar agents, as in two-period overlapping-generations models in which agents differed only as to age. Although these simple models provide insights on many issues, their limitations are becoming more apparent and we now want to move beyond them. In particular, it is desirable to add multiple goods, multiple capital stocks, heterogeneous agents, multiple assets, heterogeneous taxation, externalities, imperfect competition, and asymmetric information to the conventional models. Outside of a few special analyses, there are no examples of general dynamic, multiagent, multigood, stochastic, rational-expectations models that handle high- and low-frequency economic movements, both of which are critical for understanding data and analyzing policies. We need to turn to numerical methods to solve these models. The methods discussed in this paper move us a step in that direction.

We address the problem of computing equilibria of large rational-expectations models. We show how extensions of methods used in the simple models can be used

The authors acknowledge the comments of Wouter den Haan, two anonymous referees, and conference participants. All remaining errors are the responsibility of the authors. Dr. Judd also acknowledges the financial support of NSF grant SBR-9309613. Address correspondence to: Ken Judd, Hoover Institution, Stanford, CA 94305, USA; e-mail: judd@hoover.stanford.edu.

 1365-1005/97 $9.00 + .10

to solve more complex models. The fact that the general models considered below can be solved numerically is obvious because one can take any of the many methods that have been developed and apply them to these models. That is not an interesting observation because most of these methods' time and space requirements, such as those of discrete state-space dynamic programming, would make them impractical to solve on any computer. The problem is to develop methods that can solve larger models in reasonable time using accessible resources.

This paper focuses on the application of both perturbation and projection methods to multidimensional dynamic models. In previous papers, Judd and Guu (1993, forthcoming) examined perturbation methods that go beyond the standard linearization method, and Judd (1992) examined projection methods for solving rational-expectations models. Both papers focused on applications to a representative-agent, single-good model. The reader is referred to these papers and their mathematical sources for key definitions and introductions to these methods. In this paper, we outline how these methods can be adapted to handle multiagent generalizations; in the case of projection methods, we stay with the single-good, single-asset assumptions of the representative-agent model, but the perturbation analysis includes multiple goods and multiple capital stocks. A theme of Judd (1992) is that there are many ways to solve rational-expectations models, the best way depending on a variety of considerations. We continue that theme by attempting to determine the relationship between the most efficient method and the characteristics of the problem. We give examples of various methods, their time and space needs, and the accuracy of the results.

Our results point to a "production possibilities frontier" of methods for solving rational-expectations models. If the model is stable with an ergodic distribution concentrated in the neighborhood of some deterministic steady state, then the Taylor-series expansion produced by perturbation methods likely will produce a good approximation and, according to several existing examples, produce solutions that are reliable for a nontrivial region. If the nonlinearities are substantial and/or we need a more global approximation, we will likely use some kind of projection method.[1]

The results below give some guidance as to which methods are most efficient in terms of the trade-off between accuracy and running time. They indicate that perturbation methods are the most efficient methods when they work, and the only feasible method for the largest problems. Among the projection methods, the best choice for moderate-size models appears to use complete polynomial approximations of the unknown functions, and a version of fixed-point iteration introduced by Miranda and Helmberger (1988) to solve for the unknown coefficients. The dominance of complete polynomial representation is not surprising, but the success of fixed-point iteration is a bit surprising because it is not even locally convergent in general. However, its considerable computational advantages over Newton methods for solving nonlinear equations combined with our favorable experience (below and in earlier papers) indicates that it should generally be considered. One should keep in mind, however, that all of the models that we examine below have

strong mean reversion properties in equilibrium, a property that may help explain the success of fixed-point iteration. Because these experiments indicate that fixed-point iteration is often valuable, future work should focus on exactly when it is stable.

Although these conclusions are based on the analysis of simple examples, we feel that this experience is likely to be robust to other models and that its conclusions can serve as a reasonable guide to choosing methods for solving large models. Hopefully, the message here is encouraging enough that analysts will feel free to analyze large models of their choosing.

2. NUMERICAL RATIONAL-EXPECTATIONS PROBLEMS

Before we discuss various specific rational-expectations models, we should point out the features that are inherent in numerical rational-expectations modeling. Awareness of these features will help us to form reasonable modeling goals and point in the direction of appropriate numerical methods.

In theory, rational-expectations models resemble dynamic interpretations of Arrow–Debreu general-equilibrium analysis. The central focus of Arrow–Debreu models are the time- and state-contingent equilibrium prices and consumption patterns. The determination of equilibrium prices is the focus of standard CGE methods. However, this is an unreasonable numerical approach in the case of rational-expectations models. In the Arrow–Debreu approach to infinite-horizon dynamic stochastic models, the price of each good is contingent on the date at which it appears and the entire history of the exogenous shocks up to that date. In the case of discrete-time and infinite-horizon, if there are exogenous shocks in each period and the support of these shocks contains at least two points, the number of distinct histories in an infinite-horizon model has the cardinality of the continuum, implying that the number of distinct prices is also the size of the continuum and that the conventional CGE approach is infeasible.

These problems can be avoided if the structural features of the economic model are stationary, that is, do not depend on calendar time, and if we focus on stationary rational-expectations equilibria of those stationary infinite-horizon models. Such equilibria can be expressed in terms of decision rules for the agents [see Stokey and Lucas (1989) for a formal discussion of recursive modeling], and the numerical approach is to compute approximations of these unknown functions. For example, a simple discrete-time problem explored in the computational literature is

$$\begin{aligned} \max_c E\left\{\sum_{t=0}^{\infty}\beta^t u(c_t)\right\}, \\ k_{t+1} = F(k_t, \theta_t) - c_t, \\ \ln \theta_{t+1} = \rho \ln \theta_t + \epsilon_{t+1}, \end{aligned} \tag{1}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

where k_t is the beginning-of-period capital stock, θ_t is a stationary AR(1) multiplicative productivity parameter [the productivity shocks $\epsilon_t \sim N(0, \sigma^2)$ are independent], and $F(k, \theta)$ is the gross (i.e., net production plus initial capital stock) production function. In this problem, both k and θ are needed for a sufficient description of the state. Hence, consumption is a function of both k and θ, $C(k, \theta)$, and the Euler equation is

$$u'(C(k,\theta)) = \beta E\{u(C(F(k,\theta) - C(k,\theta), \tilde{\theta}))F_1(F(k,\theta) - C(k,\theta), \tilde{\theta}) \mid \theta)\}. \tag{2}$$

This is the problem investigated in the Taylor–Uhlig (1990) symposium and by Judd (1992). Although we focus on generalizations of (2), our results apply also to the commodity problems studied in earlier computational rational-expectations analyses by Gustafson (1958), Wright and Williams (1984), Miranda and Helmberger (1988), and Williams and Wright (1991).

At first, focusing on the functional equation in (2) appears to be no better because we move from the infinite-dimensional space of contingent prices to the infinite-dimensional space of decision rules. There is improvement, however, because there are good finite-dimensional approximations of the equilibrium policy functions, such as $C(k, \theta)$ when they are reasonable functions, whereas we know of no such way to approximate the continuum of contingent prices. Numerical rational-expectations methods, beginning with Gustafson (1958), therefore focus on finite-dimensional approximations of policy functions and other important functions. The finite-dimensional approximations typically parameterize the unknown policy function[2] and restrict it to lie in some finite-dimensional space, as in

$$\hat{C}(k,\theta) = \sum_{i=0}^{n} a_i \phi_i(k,\theta),$$

where the ϕ_i comprise a basis for all candidate functions. We first must make a choice of basis, and then solve for the unknown coefficients, a. There are several ways to fix these unknown coefficients. Projection methods fix a by solving a set of projection equations that try to identify a, which will nearly solve (2). Perturbation methods construct a Taylor-series expansion around a point (k_0, θ_0), such as

$$\hat{C}(k,\theta) = \sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij}(k-k_0)^i(\theta-\theta_0)^j,$$

and use implicit-function theorems to identify the undetermined a_{ij} coefficients.

The various approaches to solving rational-expectations models differ in three basic ways: first, in the choice of finite-dimensional approximations to functions; second, in the way the expectation in (2) is computed; and, third, in the method used to find an approximate solution. The work discussed below touches on two of the three critical elements—the method used to approximate C and the method for solving the identifying conditions. We focus on various combinations

of approximation and solution methods that appear to be promising in the context of large rational-expectations models.

We discuss both perturbation and projection methods. They share much in common; see Judd (1996) for an extended discussion of this observation. We focus on applications of perturbation methods to continuous-time models and applications of projection methods to discrete-time models. The dynamic economics literature bounces between continuous- and discrete-time applications, depending on whether one builds on the Brock and Turnovsky (1981) analysis of dynamic macroeconomic equilibrium, or the Brock and Mirman (1972) stochastic growth model. There is no substantive economic difference because the discrete-time unit can be made arbitrarily small. We make our choices to focus on the simple cases; both methods are applicable to both types of models, but perturbation analysis of discrete-time models is much more complex notationally, and projection analysis of continuous-time models would require us to introduce functional analytic material of little interest to the intended reader. We leave these developments to future work.

3. PERTURBATION METHODS

We first explore an example of perturbation methods applied to the canonical continuous-time, stable optimal control problem. We do not define perturbation in general here, nor do we present the mathematical foundations that justify the formal, algebraic manipulations presented below; see Judd (1996, forthcoming) and Bensoussan (1988) for general discussions and for citations of the formal development of perturbation methods for control problems. Here, we take a "linearize around the steady state" approach to large rational-expectations models and extend it, via perturbation methods, to include nonlinear terms of the Taylor-series expansion. Although the linearization approach is common, general perturbation methods have been used only to a limited extent in economics. There are several reasons for this. First, there is little agreement in the literature[3] as to what constitutes linearization. Second, even those who linearize deterministic models correctly[4] generally fail to compute the true first-order Taylor-series approximation when they approximate stochastic models; we make this precise below. Third, there is very limited mention in the economics literature of the higher-order terms in the expansion implicitly being constructed. In fact, Marcet (1994, p. 111) states that "perturbation methods of order higher than one are considerably more complicated[5] than the traditional linear-quadratic case." Below we show that whereas computing the traditional linear-quadratic approximation involves solving a Riccati-like equation, computing those higher-order terms involves solving only linear equations, and, therefore, that computing some higher-order terms is in fact less demanding computationally than computing the initial linear term.

High-order expansions are useful for two reasons. First, higher-order terms are necessary for analyzing the first-order properties of many aspects of the model; an example of this would be the cyclical properties of risk premia that clearly involve

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

how the risk-aversion coefficient moves over time, a third-order property of utility. The second reason, and the focus of this section, is that Taylor-series expansions that include nonlinear terms may provide good approximations over a larger region than the linear approximation would. The fact that much effort has been extended to compute nonlinear solutions to dynamic growth models, such as in Taylor and Uhlig (1990), indicates that we do not believe that linear approximations are always adequate.

In this section, we present the standard mathematical procedure for linearizing a model around a stable steady state,[6] for computing higher-order terms in the expansion, and for computing the extra terms necessary for an asymptotically valid expansion in stochastic models. We break down the method into the relevant computational steps, and evaluate the computational cost of large models. We focus on the mechanics of computing Taylor-series expansions for large, general dynamic programming problems and the associated computational demands. Future work will examine the quality of the approximation in various specific contexts, and on generalizing the analysis to the case of competitive equilibrium.

Before continuing, we should warn the reader of the nontrivial notational challenge that awaits him or her in the sections below. After being introduced to tensor notation and its application to multivariate stochastic control, the reader may decide that this approach is far too burdensome to be of value. If one had to go through these manipulations for each and every application, we might agree. Fortunately, all of the algebra discussed below can be automated. Furthermore, the authors are writing user-friendly programs that will take specifications of tastes and technology (represented by some user-written subroutines) and will automatically perform *all* necessary computations, including derivatives, and produce the Taylor-series approximation discussed below. This will relieve the user of executing all of the algebra we discuss below. This software will be available on a public archive; the interested reader should consult the second author's web page at http://www-hoover.stanford.edu/bios/judd.html or contact him at judd@hoover.stanford.edu for directions. The presentation below is meant to familiarize the reader with the mathematical structure of the problem. To help interested readers better understand the algebraic details, we have also worked out a two-dimensional example in detail; that will also be available with the programs.

3.1. Uses of Taylor-Series Approximations

To motivate the following computations, we next indicate what we can do with these Taylor-series approximations. First, the focus of this paper, we can use them to serve as global (or, more precisely, nonlocal) approximations. This may seem inappropriate because the Taylor-series construction is only local.[7] However, the global accuracy experiments conducted so far [e.g., see Judd and Guu (1993, forthcoming)] indicate that these local methods do well in nontrivial neighborhoods of the steady state in dynamic economic problems as long as the utility and production functions are analytic in a neighborhood of the deterministic steady state. Second,

we can produce linear theories of any interesting economic phenomenon.[8] Linear approximations of policy functions will produce linear theories about equilibrium choices, such as consumption, investment, and output. However, if we want to produce a linear theory for other aspects of equilibrium, we often will need higher-order terms. A linear theory for the movement of the equity risk premium over the business cycle, or a similar linear theory of the term structure of interest rates, requires higher-order Taylor expansions of the equilibrium decision rules.

Third, these approximations are the first step in potential empirical procedures because a linear theory of an economic quantity can be used to compare theory and data. For example, Magill (1977) showed how to use linear approximations of a stochastic growth model to compute the implied spectrum for consumption, and suggested that comparing such theoretical spectra with empirical spectra can be a useful empirical approach to business-cycle investigations. Later work by Kydland and Prescott (1982) used Magill's approach to compare data and a particular stochastic growth model. The Taylor series we compute below can be similarly used to compare theory and data because these expansions also can be used to compute locally valid approximations of likelihood functions and their derivatives. Fourth, these perturbation methods have been used in policy evaluation exercises. See Judd (1996) for a more extensive discussion of these topics.

3.2. General Dynamic Optimization Problem

For the purposes of this discussion, it is advantageous to discuss the continuous-time dynamic programming case.[9] We could use the equivalence between competitive equilibrium and Pareto efficiency to derive approximations to competitive equilibrium decision rules. These methods can handle distortions, but with some added complexity that the authors leave for future work. In general, if we have many agents, many goods, and many capital stocks, including "stocks" in the utility functions, the dynamic optimization problem is

$$\begin{aligned} V(x_0) &\equiv \max_{u(t)} \int_0^\infty e^{-\rho t}\pi(x,u)\,dt, \\ \dot{x} &= f(x,u), \\ x(0) &= x_0, \end{aligned} \tag{3}$$

where $x \in R^n$ is the state vector, $u \in R^m$ is the vector of controls, $f(x,u)$ is the law of motion, and $\pi(x,u)$ is a concave social welfare function. This problem has the Hamiltonian $H(x,\lambda,u) = \pi(x,u) + \lambda f(x,u)$, implying the differential–algebraic system

$$\begin{aligned} \dot{x} &= f(x,u) = H_\lambda, \\ \dot{\lambda} &= \rho\lambda - (\lambda f_x(x,u) + \pi_x) = \rho\lambda - H_x, \\ 0 &= \pi_u(x,u) + \lambda f_u(x,u) = H_u, \end{aligned} \tag{4}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

When we substitute the control law $\mathcal{U}(x, \lambda)$, implicitly defined by $0 = \pi_u(x, \mathcal{U}(x, \lambda)) + \lambda f_u(x, \mathcal{U}(x, \lambda))$, into (4), we arrive at the dynamic system

$$\begin{aligned} \dot{x} &= f(x, \mathcal{U}(x, \lambda)), \\ \dot{\lambda} &= \rho\lambda - (\lambda f_x(x, \mathcal{U}(x, \lambda)) + \pi_x(x, \mathcal{U}(x, \lambda))). \end{aligned} \tag{5}$$

Furthermore, we are interested only in asymptotically bounded solutions to this equation with the initial condition $x(0) = x_0$. This is the typical kind of deterministic dynamic system with which we begin in our linearization exercises.

3.3. Local Dynamics

To linearize systems such as (5), one invokes basic ordinary differential equation theory. Let

$$Z = \begin{pmatrix} x \\ \lambda \end{pmatrix}.$$

Suppose (5) is a dynamic system of the form

$$\dot{Z} = g(Z) \tag{6}$$

with a stationary point at Z^*; that is, $g(Z^*) = 0$. Then, the local behavior of (6) for Z near Z^* is linearly approximated by the linear system

$$\dot{z} = Az, \tag{7}$$

where $A = g_Z(Z^*)$ and $z \equiv Z - Z^*$. The solution to (7) is $z(t) = e^{At}z_0$.

In the terminology of linear rational-expectations models, as in Blanchard and Kahn (1980), the vector x contains the predetermined variables and λ contains the variables with free values at $t = 0$. Suppose that there is a stationary point at

$$Z^* = \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix}.$$

Then the local behavior of the system is linearly approximated by (7) and the solution to the linear approximation is

$$z(t) = e^{At} \begin{pmatrix} x(0) - x^* \\ \lambda(0) - \lambda^* \end{pmatrix},$$

where $x(0) = x_0$ is a given initial condition and $\lambda(0)$ is chosen to keep $z(t)$ bounded asymptotically. Let $\Lambda(x_0)$ be the set of all possible values for the free variables in λ which together with the predetermined variables being equal to x_0 will imply a bounded path for $z(t)$; $\Lambda(x_0)$ may be a single value or a set of values. We assume that it is single-valued, the case of determinacy. We show later where that is necessary in our calculations.

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

To compute $\Lambda(x_0)$, we just apply standard linear algebra. We form the Jordan decomposition of $A = N^{-1}DN$, where

$$D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}$$

with D_1 having all the stable eigenvalues of A (i.e., the eigenvalues with negative real parts) and D_2 having the unstable eigenvalues, and

$$N = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix}$$

breaks N into equal-size blocks. If the number of stable eigenvalues equals the length of x_0, stability of the solution to (7) implies

$$\Lambda(x_0) - \lambda^* = -N_{22}^{-1}N_{21}(x_0 - x^*). \tag{8}$$

We can apply this same approach to discrete-time systems. Suppose that we have a system, $Z_{t+1} = g(Z_t)$, with a steady state defined by $Z^* = g(Z^*)$. Define $A = g_Z(Z^*)$ and $z \equiv Z - Z^*$. Then, the equation for z becomes $z_{t+1} = Az_t$, the solution of which is $z_t = A^t z_0$, which in turn can be analyzed in the same Jordan decomposition fashion with the distinction that now the stable eigenvalues are those with modulus less than one. Unfortunately, the general discrete-time case is more complex, and we leave that to future work.

3.4. Higher-Order Approximations

We next discuss the computational demands of computing higher-order terms to the multivariate Taylor-series approximation based at the deterministic steady state. This may seem formidable. To solve the first-order terms we had to solve an eigenvalue-eigenvector problem. In fact, if one were to look at the details, one would find that the problem is very similar to a Riccati equation, that is, a quadratic matrix equation. When discussing this problem with macroeconomists, we often have heard the conjecture that computing higher-order terms would require solving higher-order matrix polynomial problems. The basic fact shown below is that all higher-order terms of the Taylor-series expansion, even in the stochastic multidimensional case, are solutions to linear problems once one computes the first-order terms. This implies that the higher-order terms are *easy* to compute. Initial experiments indicate that they are also good approximations well beyond the steady-state values. These procedures have not been exploited, but obviously can be applied to problems in the real business cycle, finance, public finance, and dynamic general equilibrium literatures.

3.4.1. Tensor notation. To deal with the notational problems that arise with multidimensional expressions, we extend the Einstein tensor notation, adapting it

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

for our control theoretic problems. In general, a tensor is any indexed collection of numbers. The second property is that we use the position and repetition of indices to indicate summation. Suppose that a_i is a collection of numbers indexed by $i = 1, \ldots, n$, and that x^i is also a singly indexed collection of real numbers. Then,

$$a_i x^i \equiv \sum_i a_i x^i.$$

In this way we eliminate the $\sum$ symbol in expressing vector products. Similarly, suppose that a_{ij} is a collection of numbers indexed by $i, j = 1, n$, and that x^i and y^j are singly indexed collections of real numbers. Then,

$$a_{ij} x^i y^j \equiv \sum_i \sum_j a_{ij} x^i y^j.$$

In this way we again eliminate the $\sum$ symbols. The general rule is that we eliminate $\sum$ symbols by understanding that, in a product, if an index appears as both a subscript and a superscript, then we sum over it. If we think of a^i_j as a matrix, x_i as a row vector, and y^j as a column vector, then the product $x_i y^i$ represents the inner product of the vectors x and y, and $a^i_j x_i y^j$ is the quadratic form of the matrix a with the vectors x and y. We also can form new indexed collections of numbers from products. For example, the product $a^i_j x_i$ can be thought of as a singly indexed collection of numbers, z_j. Using our vector analogy above, z_j also can be thought of as a row vector. Although the analogies with matrices and vectors are useful, one should not focus on them because we will be constructing more complex collections of real numbers that are neither vectors nor matrices.

As long as indices are not the same, arbitrary products are allowed. For example, $x_i y_j$ is the doubly indexed set of numbers, b_{ij}, where the (i, j) term equals the product of x_i and y_j; b_{ij} is the *outer product* of x_i and y_j. Also, the "sum over repeated indices" rule even applies within a single term. For example, $a^i_i \equiv \sum_i a^i_i$; if we think of a as a matrix, then a^i_i is the trace of a. Similarly, a_{ii} is also the trace of the tensor a_{ij}.

We also vary the notation to distinguish between an argument of a vector and a derivative, and between states and controls in an efficient fashion. In the notation below, superscripts refer to different components of a vector-valued function, whereas subscripts refer to derivatives of those component functions. Furthermore, to distinguish between states and controls, we let Roman letters, $i, j, k, \ell, \ldots$, index states, and Greek letters, $\alpha, \beta, \gamma, \ldots$, index controls. Therefore, if $f(x, u)$ is a vector-valued function of the state variables x and the controls u, then f is the column vector

$$f = \begin{pmatrix} f^1(x, u) \\ f^2(x, u) \\ \vdots \\ f^n(x, u) \end{pmatrix}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

whose ith component function is denoted f^i. Its derivatives with respect to the state variables are represented by the tensor

$$f_i^j(x,u) \equiv \frac{\partial f^j}{\partial x_i}(x,u),$$

and its derivatives with respect to the control variables are represented by the tensor

$$f_\alpha^j(x,u) \equiv \frac{\partial f^j}{\partial u_\alpha}(x,u).$$

We frequently drop the argument (x,u) as long as it can be understood from context.

So far, all of this looks familiar because it is just a way of rewriting standard matrix and vector operations. However, triply indexed collections of numbers, such as $a_{ij\ell}$, are also tensors, and arise naturally in multivariate calculus. For example, Taylor's theorem for $g\colon R^n \to R$ at $z = 0$ is normally written as

$$f(z) \sim f(0) + \sum_{i=1}^{n} \frac{\partial f}{\partial z_i}(0)\, z_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial z_i \partial z_j}(0)\, z_i z_j + \frac{1}{6}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\ell=1}^{n} \frac{\partial^3 f}{\partial z_i \partial z_j \partial z_\ell}(0)\, z_i z_j z_\ell + \cdots,$$

whereas by using tensor notation it can be written as

$$f(z) \sim f(0) + f_i(0)z^i + \tfrac{1}{2} f_{ij}(0)z^i z^j + \tfrac{1}{6} f_{ij\ell}(0)z^i z^j z^\ell + \cdots.$$

If we drop the arguments of f, and understand that f and its derivatives are evaluated at $x = 0$, the Taylor expansion can be written as

$$f(z) \sim f + f_i z^i + \tfrac{1}{2} f_{ij} z^i z^j + \tfrac{1}{6} f_{ij\ell} z^i z^j z^\ell + \cdots.$$

This more compact expression is a considerable improvement over the conventional notation with the extraneous summation symbols and the clumsy partial derivative notation. With this notation, we are able to see more clearly the structure of our problem.

3.4.2. Computing Taylor-series expansions for optimal control problems. The general deterministic problem (3) includes an arbitrary complete market dynamic equilibrium with multiple consumption goods (represented by components of u), multiple capital stocks (represented by components of x), and multiple agents whose aggregate utility function is represented by $\pi(x,u)$. We now proceed with a dynamic programming approach to this analysis. The Bellman equation for the value function, $V(x)$, is

$$0 = \max_u \pi(x,u) + V_i(x) f^i(x,u) - \rho V(x), \tag{9}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

where, recall, π is the payoff flow and f is the law of motion for the state x. The first-order condition with respect to u^α, $\alpha = 1, \ldots, m$, is

$$0 = \pi_\alpha(x,u) + V_i(x)\, f^i_\alpha(x,u). \tag{10}$$

Equations (10) implicitly define the optimal control, $u = U(x)$, and imply the system

$$0 = \pi_\alpha(x,U(x)) + V_i(x)\, f^i_\alpha(x,U(x)). \tag{11}$$

In combination, we have the system

$$0 = \pi(x,U(x)) + V_i(x)\, f^i(x,U(x)) - \rho V(x), \tag{12}$$

$$0 = \pi_\alpha(x,U(x)) + V_i(x)\, f^i_\alpha(x,U(x)), \tag{13}$$

which defines the value function, $V(x)$, and the policy function, $U(x)$.

Our objective here is to solve for both the value function and the policy function. In fact, we are going to compute Taylor-series expansions

$$\begin{aligned} V(x) &= V(x^0) + V_i(x-x^0)^i + \tfrac{1}{2}V_{ij}(x-x^0)^i(x-x^0)^j \\ &\quad + \tfrac{1}{3!}V_{ij\ell}(x-x^0)^i(x-x^0)^j(x-x^0)^\ell + \cdots, \\ U^\alpha(x) &= U^\alpha(x^0) + U^\alpha_i(x-x^0)^i + \tfrac{1}{2}U^\alpha_{ij}(x-x^0)^i(x-x^0)^j \\ &\quad + \tfrac{1}{3!}U^\alpha_{ij\ell}(x-x^0)^i(x-x^0)^j(x-x^0)^\ell + \cdots. \end{aligned} \tag{14}$$

These polynomials are asymptotically valid approximations if the error converges to zero at a higher degree than the order of the polynomial. For example, the linear approximation $V(x^0) + V_i(x-x^0)^i$ has an error that is quadratic in the components of $(x-x^0)$, which means that, as x converges to x^0, the error goes to zero quadratically.

To compute these U^α_i, U^α_{ij}, $U^\alpha_{ij\ell}$, V_i, V_{ij}, and $V_{ij\ell}$ coefficients, we just differentiate the underlying system (12, 13), with respect to the x_i, and solve for the undetermined coefficients. If we differentiate (12) with respect to x_j and use the envelope theorem, we find

$$\rho V_j = \pi_j + V_{ij}\, f^i + V_i\, f^i_j. \tag{15}$$

To keep down the clutter, we often drop the arguments of π, V, and U, and their derivatives when they are the same as in the basic system (12, 13) and are clear from context.

The steady-state values for x, u, and V_i are determined by the conditions

$$\begin{aligned} 0 &= f^i(u,x), \\ 0 &= \pi_\alpha(u,x) + V_i(x)\, f^i_\alpha(u,x), \\ \rho V_j(x) &= \pi_j(u,x) + V_{ij}(x)\, f^i(u,x) + V_i(x)\, f^i_j(u,x), \end{aligned} \tag{16}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

which yield the steady-state quantities $u^{ss}, x^{ss}, V_j(x^{ss})$. Note that the $V_j(x^{ss})$ values are the linear coefficients in the expansion of V in (14), and knowing the steady state will also yield $V(x^{ss})$ and $U(x^{ss})$, two more terms in (14).

We next compute the V_{ij} and U_j^β terms, and then compute many high-order derivatives. We assume that these derivatives exist, which leads us to assume that all production and utility functions are C^∞. We *assume* the differentiability of the value and policy functions in the neighborhood of the deterministic steady state. For deterministic problems, this is usually proven by applying theorems about the smooth dependence of differential equation solutions on parameters, and Fleming (1971) deals with the stochastic problem. We proceed under the assumption that the indicated derivatives exist.

If we differentiate (13) with respect to the x_j, we find

$$0 = \pi_{\alpha j} + \pi_{\alpha\gamma} U_j^\gamma + V_{ij}\, f_\alpha^i + V_i \left(f_{\alpha j}^i + f_{\alpha\gamma}^i U_j^\gamma \right). \tag{17}$$

Note that (17) is a system of conditions, one for each αj pair. In this case, we can express U_j^γ in terms of the derivatives of V:

$$U_j^\gamma = -\left(\pi_{\alpha\gamma} + V_i\, f_{\alpha\gamma}^i\right)^{-1} \left(\pi_{\alpha j} + V_{ij}\, f_\alpha^i + V_i\, f_{\alpha j}^i\right), \tag{18}$$

where $(\pi_{\alpha\gamma} + V_i\, f_{\alpha\gamma}^i)^{-1}$ denotes the inverse tensor (matrix). Differentiating (15) with respect to x_ℓ implies

$$\rho V_{j\ell} = \pi_{j\ell} + \pi_{j\gamma} U_\ell^\gamma + V_{ij\ell}\, f^i + V_{ij} \left(f_\ell^i + f_\gamma^i U_\ell^\gamma \right) + V_{i\ell}\, f_j^i + V_i \left(f_{j\ell}^i + f_{j\gamma}^i U_\ell^\gamma \right). \tag{19}$$

Substituting (18) into (19) yields

$$\begin{aligned} \rho V_{j\ell} &= \pi_{j\ell} + V_{ij\ell}\, f^i + V_{ij}\, f_\ell^i + V_{i\ell}\, f_j^i + V_i\, f_{j\ell}^i \\ &\quad - \left(\pi_{j\gamma} + V_{ij}\, f_\gamma^i + V_i\, f_{j\gamma}^i\right) \left(\pi_{\alpha\gamma} + V_i\, f_{\alpha\gamma}^i\right)^{-1} \left(\pi_{\alpha\ell} + V_{i\ell}\, f_\alpha^i + V_i\, f_{\alpha\ell}^i\right). \end{aligned} \tag{20}$$

The system of equations in (20) hold at each state x. If we evaluate (20) at the steady state, then $f^i = 0$ and (20) becomes the Riccati-like equation

$$\begin{aligned} \rho V_{j\ell} &= \pi_{j\ell} + V_{ij}\, f_\ell^i + V_{i\ell}\, f_j^i + V_i\, f_{j\ell}^i \\ &\quad - \left(\pi_{j\gamma} + V_{ij}\, f_\gamma^i + V_i\, f_{j\gamma}^i\right) \left(\pi_{\alpha\gamma} + V_i\, f_{\alpha\gamma}^i\right)^{-1} \left(\pi_{\alpha\ell} + V_{i\ell}\, f_\alpha^i + V_i\, f_{\alpha\ell}^i\right). \end{aligned} \tag{21}$$

Solving the Riccati equation at the steady state yields the steady-state values of $V_{j\ell}$ and, through (18), the steady-state values of U_ℓ^γ. However, we already have the solutions for the steady-state values of U_ℓ^γ because the U_ℓ^γ tensor can be derived from the definition of $\mathcal{U}(x, \lambda)$ and the linear approximation computed in (8). This approximation was computed directly from eigenvalue decomposition methods. Therefore, we use the earlier approach to compute the elements of the tensor

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

(matrix) U_ℓ^γ. Although we have not accomplished anything new at this point, this lays the foundation for the higher-order terms.

We now find out how easy it is to compute the higher-order terms. Differentiating (19) with respect to x_m and imposing the steady-state condition $f^i = 0$ implies the following equation for the steady-state values of $V_{i\ell m}$ and $U_{\ell m}^\beta$:

$$\begin{aligned}
\rho V_{j\ell m} &= \pi_{j\ell m} + \pi_{j\ell\gamma} U_m^\gamma + \pi_{j\gamma m} U_\ell^\gamma + \pi_{j\gamma\delta} U_m^\delta U_\ell^\gamma + \pi_{j\gamma} U_{\ell m}^\gamma \\
&\quad + V_{ij\ell}\left(f_m^i + f_\gamma^i U_m^\gamma\right) + V_{ijm}\left(f_\ell^i + f_\gamma^i U_\ell^\gamma\right) \\
&\quad + V_{ij}\left(f_{\ell m}^i + f_{\ell\gamma}^i U_m^\gamma + f_{\gamma m}^i U_\ell^\gamma + f_{\gamma\delta}^i U_\ell^\gamma U_m^\delta + f_\gamma^i U_{\ell m}^\gamma\right) \\
&\quad + V_{i\ell m} f_j^i + V_{i\ell}\left(f_{j\gamma}^i U_m^\gamma + f_{jm}^i\right) + V_{im}\left(f_{j\ell}^i + f_{j\gamma}^i U_\ell^\gamma\right) \\
&\quad + V_i\left(f_{j\ell m}^i + f_{j\ell\gamma}^i U_m^\gamma + f_{j\gamma m}^i U_\ell^\gamma + f_{j\gamma\delta}^i U_\ell^\gamma U_m^\delta + f_{j\gamma}^i U_{\ell m}^\gamma\right). \qquad \textbf{(22)}
\end{aligned}$$

When we rewrite (17) as

$$0 = \left(\pi_{\alpha\beta} + V_i f_{\alpha\beta}^i\right) U_\ell^\beta + \left(\pi_{\alpha\ell} + V_{i\ell} f_\alpha^i + V_i f_{\alpha\ell}^i\right)$$

and differentiate this expression with respect to x_m, and impose the steady-state condition, we find

$$\begin{aligned}
0 &= \left(\pi_{\alpha\beta} + V_i f_{\alpha\beta}^i\right) U_{\ell m}^\beta \\
&\quad + \left(\pi_{\alpha\beta\gamma} U_m^\gamma + \pi_{\alpha\beta m} + V_{im} f_{\alpha\beta}^i + V_i f_{\alpha\beta m}^i + V_i f_{\alpha\beta\gamma}^i U_m^\gamma\right) U_\ell^\beta \\
&\quad + \pi_{\alpha\ell m} + \pi_{\alpha\ell\gamma} U_m^\gamma + V_{i\ell m} f_\alpha^i + V_{i\ell}\left(f_{\alpha m}^i + f_{\alpha\gamma}^i U_m^\gamma\right) \\
&\quad + V_{im} f_{\alpha\ell}^i + V_i\left(f_{\alpha\ell m}^i + f_{\alpha\ell\gamma}^i U_m^\gamma\right). \qquad \textbf{(23)}
\end{aligned}$$

We have now reached an important point in the analysis. At this point, we know the steady-state values of the U_ℓ^β and V_{im} tensors as well as the steady-state values of the derivatives of π and f that appear. We see that the steady-state values of $V_{i\ell m}$ and $U_{\ell m}^\beta$ appear *linearly* in (22), (23) and that they are the only unknowns. The system is not cubic or quadratic, but linear in these unknowns.

The second important point is that solving this linear system is easier than it initially appears. An efficient way to compute the steady-state values of the $V_{i\ell m}$ and $U_{\ell m}^\beta$ in (22), (23) is immediately apparent. The key observation is that we can solve the system for each fixed ℓm pair, allowing us to break down the large system into a collection of smaller ones. Consider (23) for a fixed ℓm pair. The $\pi_{\alpha\beta} + V_i f_{\alpha\beta}^i$ tensor appears repeatedly for each ℓm pair. Hence, we can use the $(\pi_{\alpha\beta} + V_i f_{\alpha\beta}^i)^{-1}$ tensor (an inversion that is done just once and used for each ℓm pair) to express $U_{\ell m}^\beta$ linearly in terms of the known steady-state values of various steady-state derivatives and $V_{j\ell m}$. After we gather terms, it takes just two

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

matrix multiplications for each ℓm pair to determine the coefficients of the affine representation

$$\begin{aligned} U^{\beta}_{\ell m} &= -\left(\pi_{\alpha\beta} + V_i\, f^i_{\alpha\beta}\right)^{-1} \\ &\quad \times \left[\left(\pi_{\alpha\beta\gamma} U^{\gamma}_{m} + \pi_{\alpha\beta m} + V_{im}\, f^i_{\alpha\beta} + V_i\, f^i_{\alpha\beta m} + V_i\, f^i_{\alpha\beta\gamma} U^{\gamma}_{m}\right) U^{\beta}_{\ell}\right. \\ &\quad \left. + \pi_{\alpha\ell m} + \pi_{\alpha\ell\gamma} U^{\gamma}_{m} + V_{i\ell}\left(f^i_{\alpha m} + f^i_{\alpha\gamma} U^{\gamma}_{m}\right) + V_{im}\, f^i_{\alpha\ell} + V_i\left(f^i_{\alpha\ell m} + f^i_{\alpha\ell\gamma} U^{\gamma}_{m}\right)\right] \\ &\quad - \left(\pi_{\alpha\beta} + V_i\, f^i_{\alpha\beta}\right)^{-1} f^i_{\alpha} V_{i\ell m}. \end{aligned} \tag{24}$$

For a fixed ℓm pair, these representations of the $U^{\beta}_{\ell m}$ then can be substituted into the steady-state value of (22) to produce a system of equations linear in the $V_{j\ell m}$. Hence, we see that the second-order terms can be computed in a sequentially *linear* fashion once the first-order terms have been computed. If there are n states and m controls, the total computational burden is the $2n \times 2n$ eigenvalue–eigenvector decomposition, one $m \times m$ inversion, $m(m-1)/2$ evaluations of the expression in (24), and solving a linear system of size $n(n-1)(n-2)/6$.

The final important fact is that we can repeat this to compute the third- and higher-order terms of U in a similar sequentially linear fashion. We suspect that the reader will believe us on this point without seeing the algebraic details. We may ask how big can we go in this fashion. Clearly, the greater the dimension, the fewer higher-order terms we can add to the expansion. We investigate feasibility issues below.

This may appear confusing and surprising because standard intuition says that higher-order terms are more difficult to compute. The basic explanation is that the initial linearization step must deal with the multiplicity that arises due to the steady state lying on both the stable and the unstable manifolds, a multiplicity manifested by the multiple solutions to the Riccati-like equation that arises. This multiplicity arises because our methods use only the first-order conditions, conditions that describe the unstable and the stable manifolds as well as many other manifolds. Once the linear expansion term makes a choice of which manifold to follow, a task accomplished in (8), we have fixed our attention on the stable manifold, and higher-order terms involve only easy linear problems.

3.4.3. Multisector, stochastic growth. We next examine the stochastic generalization of (3), which is

$$\begin{aligned} \max_{u} \quad & E\left\{\int_0^{\infty} e^{-\rho t}\pi(x,u)\,dt\right\}, \\ & dx^i = f^i(x,u)\,dt + \sqrt{2\epsilon}I\,dz, \end{aligned} \tag{25}$$

where dz is a vector of i.i.d. white noises of unit variance, I is the variance–covariance matrix, assumed here to be the identity matrix to reduce the notational

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

burden, and ϵ is a parameter expressing the absolute magnitude of the variances. Again, any dynamic, complete market general equilibrium can be represented by the solution to such a problem. The solutions to this problem can be used to represent consumption allocation processes, investment and consumption processes, as well as asset price processes.

The Bellman equation to (25) is

$$0 = \max_u \pi(x, u) + V_i f^i + \epsilon V_{ii} - \rho V, \tag{26}$$

and the first-order condition is again

$$0 = \pi_\alpha + V_i f^i_\alpha. \tag{27}$$

Again, we are trying to compute Taylor-series approximations to V and U, but here we take into account not only deviations of x from the deterministic steady state $x^0 - x$, but also deviations from the deterministic case, as measured by ϵ. Here the expansions are

$$\begin{aligned} V(x,\epsilon) &= V(x^{ss},0) + V_i(x^{ss},0)(x-x^0)^i + V_\epsilon(x^{ss},0)\epsilon \\ &\quad + V_{i\epsilon}(x^{ss},0)(x-x^0)^i\epsilon + \tfrac{1}{2}V_{ij}(x^{ss},0)(x-x^0)^i(x-x^0)^j \\ &\quad + \tfrac{1}{2}V_{\epsilon\epsilon}(x^{ss},0)\epsilon^2 + \cdots, \end{aligned} \tag{28}$$

$$\begin{aligned} U^\alpha(x,\epsilon) &= U^\alpha(x^{ss},0) + U^\alpha_i(x^{ss},0)(x-x^0)^i + U^\alpha_\epsilon(x^{ss},0)\epsilon \\ &\quad + U^\alpha_{i\epsilon}(x^{ss},0)(x-x^0)^i\epsilon + \tfrac{1}{2}U^\alpha_{ij}(x^{ss},0)(x-x^0)^i(x-x^0)^j \\ &\quad + \tfrac{1}{2}U^\alpha_{\epsilon\epsilon}(x^{ss},0)\epsilon^2 + \cdots, \end{aligned} \tag{29}$$

where all of the functions on the RHS are evaluated at $(x^{ss}, 0)$. The analysis of the deterministic problem produced all of the U^α_i, U^α_{ij}, $U^\alpha_{ij\ell}$, V_i, V_{ij}, and $V_{ij\ell}$ etc. coefficients. We now want to derive the $U^\alpha_{i\epsilon}$, $U^\alpha_{ij\epsilon}$, $V_{i\epsilon}$, $V_{ij\epsilon}$, $U^\alpha_{i\epsilon\epsilon}$, $U^\alpha_{ij\epsilon\epsilon}$, etc. coefficients.

Note additional terms due to the stochastic parameter ϵ. The usual approaches [such as in Kydland and Prescott (1982) and Christiano (1990)] take a linear approximation to the deterministic, $\epsilon = 0$, model, which here is $U^\alpha(x, \epsilon) = U^\alpha(x^{ss}, 0) + U^\alpha_i(x^{ss}, 0)(x^i - x^{i0})$, and then use that linear rule as an approximate solution to a stochastic version of the model. This ignores the additional $U^\alpha_\epsilon(x^{ss}, 0)\epsilon$ term that is suggested by this Taylor-series approach. Because the objective of all of these methods is to approximate the value of u at state $x \neq x^{ss}$ in a model in which $\epsilon \neq 0$ and base it on the $x = x^{ss}$ and $\epsilon = 0$ situation, that is, the steady state, the true linear approximation of this value includes the $U^\alpha_\epsilon(x^{ss}, 0)\epsilon$ term. If the model has the certainty-equivalence property, then this term is zero; otherwise, this term is part of the linear approximation of $U^\alpha(x, \epsilon)$ based at $x = x^{ss}$ and $\epsilon = 0$.

Differentiating (26) with respect to ϵ yields

$$0 = V_{i\epsilon} f^i + V_{ii} + \epsilon V_{ii\epsilon} - \rho V_\epsilon. \tag{30}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

At the steady state of the deterministic case studied in Section 3.4.2, (30) reduces to

$$0 = V_{ii} - \rho V_{\epsilon}, \tag{31}$$

which gives us our first correction, V_{ϵ}, term for variance.

We now move on to the other certainty nonequivalence correction terms. Differentiation of (30) with respect to x_j implies

$$0 = V_{ij\epsilon} f^i + V_{i\epsilon} f^i_j + V_{i\epsilon} f^i_{\alpha} U^{\alpha}_j + V_{iij} + \epsilon V_{iij\epsilon} - \rho V_{j\epsilon}, \tag{32}$$

which at the steady state becomes

$$0 = V_{i\epsilon} f^i_j + V_{i\epsilon} f^i_{\alpha} U^{\alpha}_j + V_{iij} - \rho V_{j\epsilon}, \tag{33}$$

which is an $n \times n$ linear system in the $V_{i\epsilon}$ unknowns. Differentiating (27) with respect to ϵ implies

$$0 = \pi_{\alpha\beta} U^{\beta}_{\epsilon} + V_{i\epsilon} f^i_{\alpha} + V_i f^i_{\alpha\beta} U^{\beta}_{\epsilon}, \tag{34}$$

which, at the deterministic steady state, reduces to another linear system. Continued differentiation shows that computing higher-order terms also reduces to linear systems. Again, it is easy to compute the higher-order expansion terms of the Taylor-series.

We have left out several details; in particular, we need to verify that these linear systems are nonsingular. Also, one needs proofs that the value functions are locally differentiable to the extent used in these formulas. These questions must be addressed on a problem-specific basis. If this approach leads to a singular system, then it breaks down, but that breakdown is checkable. Application of implicit-function theorems should be able to prove if the value function is locally differentiable. What we have shown is that the standard linearization method frequently used in rational-expectations analysis can be extended to include higher-order terms and can be extended to model deviations from certainty equivalence, and that doing so often will be simple numerically.

3.5. Computational Costs of Linearization and Higher-Order Terms

The computations above have succeeded in showing us the qualitative nature of the problems that need to be solved. We next ask how large can we go with this approach. To see how large we can go, we need to know the timing demands of the basic operations in the procedure. These exercises were conducted using Matlab on a 90-MHz Pentium. The analysis of an n-state, m-control optimal control problem can be broken down into the following steps:

1. Compute steady state: time $\sim \mathcal{O}[(2n + m)^3]$.
2. Compute Jacobian, A, of dynamic system at steady state: time $\sim \mathcal{O}[(2n)^2]$.
3. Compute Jordan canonical form of $A = N^{-1}DN$: time $\sim \mathcal{O}[(2n)^3]$.
4. Construct and solve the linear systems for higher-order and stochastic terms.

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

We now examine the computational cost of these steps. The computation of the steady state of an n-state, m-control problem is a nonlinear equation in $2n + m$ variables, which is not bad even for $n = m = 100$, and feasible for much larger problems. In fact, the steady-state problem is often similar to CGE problems; hence we can use the special methods available for CGE models. Presumably, this is not a difficult problem; if computing the steady state of the deterministic version is difficult, then it is unlikely that any method can produce a global solution to the stochastic version. Similarly, computing the Jacobian of the dynamic system is also presumably feasible. In general, the time demands of these steps vary greatly across problems, making it difficult for us to say anything generally. Hence, we do not consider these steps further, and instead focus on the generic features of this approach.

The critical limitations begin with the eigenvalue–eigenvector extraction problem. This is a necessary step for any asymptotically valid linearization method. The limiting factor in computing the higher-order and stochastic terms is the cost of constructing and solving the linear systems that arise. The cost of constructing the linear systems depends on the specific problem and depends on the number of nonzero derivatives that arise in the systems (22), (23), and other equations, and the manner in which one computes those derivatives. We make no attempt to estimate the cost of constructing these systems. We can give a reliable estimate of the cost of solving the resulting linear algebra problems. Table 1 contains the results of some experiments we ran to determine the cost of these steps. We took several random $n \times n$ matrices and computed eigenvalues and eigenvectors, for $n = 100, 200, 300, 400$. We next explored the timing for solving large linear systems of equations, the second operation. Table 1 also indicates the results for $n \times n$ matrices, $n = 100, 200, 500, 1{,}000$. The third pair of columns in Table 1 display the mean time for several random matrix inversions; we do not report the variation in any column because there was little variation. The entries roughly reflect the order n^3 operation count associated with solving linear systems and inverting matrices.

To determine the cost of computing various order terms in the Taylor-series expansion, we need to know the number of unknown coefficients of various orders. Table 2 indicates the size of the problem associated with various orders of the Taylor expansion for various dimensions; more precisely, it indicates the number of coefficients in the nth-order terms of the Taylor expansion of a dimension d

TABLE 1. Linear algebra times

Eigenvectors		Linear systems		Matrix inversion	
n	seconds	n	seconds	n	seconds
10	0.1	10	0.0025	10	0.05
100	2.6	100	0.17	100	0.27
200	25.5	200	0.77	200	2.4
300	104	500	18.3	300	9.0
400	230	1,000	150	400	24.4

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

TABLE 2. Taylor series

Dimension	Order	No. of coefficients
5	1	5
	2	15
	3	35
	4	70
	5	126
10	1	10
	2	55
	3	220
	4	715
20	1	20
	2	210
	3	1,540
	4	8,855
50	1	50
	2	1,275
100	1	100
	2	5,050
	3	171,700
d	n	$\frac{(n+d)!}{n!d!} - \frac{(n+d-1)!}{(n-1)!d!}$

function. Note that the bottom right entry gives the general formula for the nth-order coefficients in a d dimensional function.

These results give us good indications as to what is generally feasible. Because the linearization step involves computing a steady state, a Jacobian, and an eigenvalue–eigenvector decomposition, we see that moderately large systems can be linearized. The eigenvalue problem generally would be the most difficult, and we see here that a 10-state model would lead to a 20-dimensional eigenvector problem, one that takes under a second on a Pentium.[10] The computation of the quadratic terms is also possible. First, we compute the V_{ij} terms using (19), a linear problem with, according to Table 2, 55 variables, and which can be solved in under 0.1 seconds.[11] We then construct a linear problem involving the V_{ijm}, a problem with 220 variables, which takes roughly a second to solve. These values give us the information necessary to compute the U^{α}_{jm}, requiring 55 multiplications of 10×10 matrices. The third-order terms of U need the 715 fourth-order terms of V from its defining linear system, a problem taking about 40 seconds, and then do 715 matrix multiplications, a problem taking less than 3 seconds according to Table 1. These results also show that quadratic approximations of control laws of 100-dimensional models is also feasible. Even cubic approximations could be had with patience and enough space. Furthermore, it is clear that the certainty non-equivalence terms that we derived above also could be computed in the same time. Supercomputers could handle higher orders and/or larger models.

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

3.6. Global Accuracy

Perturbation methods produce the best possible asymptotically valid local approximations to a problem. However, we often want to use them for nonlocal approximations. We will not go into an extensive exploration of the global quality of the resulting approximations, a topic that must be investigated for specific problems and specific choices of tastes and technology. However, we should note that the existing literature is quite positive on the global quality of the resulting approximations. Judd and Guu (1993, forthcoming) have investigated this issue in simple growth models. Typically, that is, for empirically reasonable choices of tastes and technology, they find that the linear approximations do well for small but nontrivial neighborhoods of the deterministic steady state and that the quality of the approximations improves substantially as the higher-order terms are added. They also find that the certainty non-equivalence terms are important to achieve high-quality approximations for stochastic approximations. More precisely, they substitute the computed Taylor series into the defining equations and evaluate the resulting error. The resulting error for capital stocks near the steady state is often the order of machine zero, an accomplishment that few other methods can claim. Although their investigations have been limited to relatively small models, there is no reason to suspect that the performance of this approach will decay drastically as we move to larger models. In any case, any user of these methods should use some diagnostics to estimate the region where the constructed series is a good approximation.

3.7. Summary of Perturbation Methods

The exercises above show that computing Taylor-series expansions of dynamic optimization problems is feasible even for large problems, that is, problems with several states and controls. This includes problems with several agents, several sectors, several goods, and several factors. Minor modifications allow these methods to be applied to models with distortions; see Judd (forthcoming) for discussions of how to do this. The method described above is a substantial improvement and generalization of the conventional linearization method used commonly in public finance and macroeconomics, allowing for certainty non-equivalence. In general, it can be used to create a linear theory for any observable quantity including those, such as risk premia, that depend on nonquadratic properties of tastes and technology.

The main disadvantage of this method is its local nature. We next examine methods that are more global in their approach.

4. PROJECTION METHODS

Projection methods take a global approach to problems, making them applicable for a broader range of problems. However, they are much slower. Before discussing

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

the large models that we solve below, we review the basics of projection methods as applied to (1). That problem reduces to solving the functional equation

$$0 = u'(C(k,\theta)) - \beta E\{u'(C(F(k,\theta) - C(k,\theta),\tilde{\theta}))F_1(F(k,\theta) - C(k,\theta),\tilde{\theta}) \mid \theta\}$$
$$\equiv R(k,\theta;C), \tag{35}$$

where $R(k,\theta;C)$ is the Euler residual of the policy function $C(k,\theta)$.

The projection approach to solving (35) can be broken down into four components. First, as noted above, we parameterize the unknown policy function and restrict it to lie in some finite-dimensional space, as in

$$\hat{C}(k,\theta;a) = \sum_{i=0}^{n} a_i \phi_i(k,\theta),$$

where the ϕ_i comprise a basis for all candidate functions. Second, we define a numerical approximation of the residual function applied to $\hat{C}$, $R(k,\theta;\hat{C})$; we let $\hat{R}$ denote the numerical approximation of R. This primarily involves choosing a scheme for approximating the integral implicit in the conditional expectations. Third, we construct a collection of projection conditions; each is defined by some test function $\psi_\ell(k,\theta)$ relative to a weighting function $w(k,\theta)$, and ideally equals

$$\int R(k,\theta;\hat{C}(\cdot;a))\psi_\ell(k,\theta)w(k,\theta)\,dk\,d\theta$$

but is implemented by choosing a finite sample (k_j,θ_j), $j = 1,\ldots,m$, and defining

$$P_\ell(a) = \sum_{j=1}^{m} \hat{R}(k_j,\theta_j;\hat{C}(\cdot;a))\alpha_j\psi_\ell(k_j,\theta_j), \qquad \ell = 1,\ldots,n. \tag{36}$$

Fourth, we choose some method to compute the coefficient vector a that solves the system $P_\ell(a) = 0$.

The various approaches to solving rational-expectations models differ in their choices over these components. Table 3 displays the various choices. The critical fact is that this menu is "a la carte": You make one choice from column A, then a choice from column B, etc., and almost any combination is possible. Table 3 outlines the basic components of projection methods, and points out the specific choices that have been and could be used.

The existing literature on computational rational-expectations methods have pursued a small fraction of the possible combinations. Table 4 displays the various combinations that have been used. We left out the projection-condition column because almost all essentially use collocation and Galerkin-style methods.

We do not consider the integration problem here, using product Gaussian quadrature in our examples; substantial improvement is possible by developing truly multidimensional quadrature and approximation methods, but we leave that for future research. In our examples, we focus on some of the approximation choices and

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

TABLE 3. Projection method components

Approximation	Integration	Projection conditions	Solution method
Piecewise linear	Newton–Cotes	Galerkin	Newton
Ordinary polynomial	Gaussian quadrature	Collocation	Time iteration
Orthogonal polynomial	Monte Carlo	Sum of squares	Fixed-point iteration
Splines	quasi-Monte Carlo	Subdomain	Least squares
Neural networks	R^n rules	Method of moments	Global minimum
Finite element	Derivative rules		Homotopy
Customized bases	Asymptotics		

TABLE 4. Choices made in literature

Authors	Approximation	Integration	Solution method
Gustafson (1958)	Piecewise linear	Newton–Cotes	Time iteration
Wright and Williams (1982,1984)	Polynomial (of conditional expectation)	Newton–Cotes	Time iteration
Miranda and Helmberger (1988)	Polynomials	Newton–Cotes	Fixed-point iteration
Bizer and Judd (1989)	Piecewise linear	Newton–Cotes	Time iteration
Coleman (1990)	Finite element	Gaussian	Time iteration
den Haan and Marcet (1990)	Polynomial (of conditional expectation)	Monte Carlo simulation	Fixed-point iteration
Judd (1992)	Orthogonal polynomial	Gaussian	Newton

solution-method choices. Judd (1992) contains the results for various methods for the representative-agent model. We now look at extensions of this model.

5. HETEROGENEOUS TASTES

We next examine how to introduce several agents into the analysis. We examine the same model as in (1) except that we assume that there are n different types of agents, type i having utility function, $u_i(c)$, $i = 1, 2, \ldots, n$, but a common discount factor β. In this case, the equilibrium decisions depend on the distribution of wealth. Let $C^i(k)$ be the consumption of type i agents when the wealth distribution is $k = (k_1, k_2, \ldots, k_n)$. We assume that equity is the only asset that can be held. We do this to focus on the issues of size of models, whereas introducing a larger set

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

of securities introduces other difficulties. The equilibrium of the resulting model is defined by the collection of Euler equations

$$R^i(k,\theta,C) = u_i'(C^i(k,\theta)) - \beta E\{u_i'(C^i(Y(k,\theta) - C(k,\theta),\tilde{\theta}))$$
$$\times F_1((Y(k,\theta) - C(k,\theta))_\cdot,\tilde{\theta}) \mid \theta\}, \qquad i = 1,2,\ldots,n,$$

where $Y(k,\theta) \in R^n$ is the distribution of income in a period with initial capital stock distribution k and productivity θ, that is, $Y^i(k,\theta) = k_i F_1(k_\cdot,\theta) + W(k,\theta)$ and $W(k,\theta) = F(k_\cdot,\theta) - k_\cdot F_1(k_\cdot,\theta)$, where we use the tensor notation $k_\cdot \equiv \sum_i k_i$. We focus on the residual functions

$$R^i(k,\theta,\hat{C}(\cdot;a)) = \hat{C}^i(k,\theta;a) - (u_i')^{-1}(\beta E\{u_i'(\hat{C}^i(Y(k,\theta) - \hat{C}(k,\theta;a),\tilde{\theta};a))$$
$$\times F_1((Y(k,\theta) - \hat{C}(k,\theta;a))_\cdot,\tilde{\theta}) \mid \theta\}), \qquad i = 1,2,\ldots,n,$$

where we approximate each consumption function in terms of some unknown coefficients a, as in $\hat{C}^i(k,\theta;a)$. Because of the presence of the expectation operator, we need to form the approximate residual function for agent i,

$$\hat{R}^i(k,\theta,\hat{C}(\cdot;a)) = \hat{C}^i(k,\theta;a) - (u_i')^{-1}(\beta \hat{E}\{u_i'(\hat{C}^i(Y(k,\theta;a) - \hat{C}(k,\theta;a),\tilde{\theta};a))$$
$$\times F_1((Y(k,\theta) - \hat{C}(k,\theta;a))_\cdot,\tilde{\theta}) \mid \theta\}), \qquad i = 1,2,\ldots,n,$$

where $\hat{E}$ represents some numerical approximation of the enclosed integral. In this paper we use only product Gaussian quadrature formulas for integration. The identifying projections are

$$P_{ij}(a) \equiv \int_{\theta_m}^{\theta_M}\int_{k_m}^{k_M}\cdots\int_{k_m}^{k_M} \hat{R}^i(k,\theta,\hat{C}(\cdot;a))\psi_j(k,\theta)w(k,\theta)\,dk_1\cdots dk_n\,d\theta,$$

where $i = 1,\ldots,n$, and $j = 1,\ldots,m$. The computation of $P(a)$ also involves numerical integration; we let $\hat{P}(a)$ denote a numerical integration approximation of $P(a)$. We use product Gaussian quadrature here also; we leave the integration issues for future research. The solution chooses a so that $\hat{P}(a) = 0$. We next discuss the leading possibilities for the approximation scheme and the solution method.

5.1. Representation: Tensor vs. Complete Polynomials

The tensor method approximates each consumption function as

$$\hat{C}^i(k,\theta;a) = \sum_{j_1=0}^{n_k}\cdots\sum_{j_n=0}^{n_k}\sum_{\ell=0}^{n_\theta} a^i_{j_1\cdots j_n\ell}\varphi_{i_1}(k_1)\cdots\varphi_{i_n}(k_n)\psi_\ell(\theta), \qquad i = 1,\ldots,n,$$

where $\varphi_i(k_j)(\psi_\ell(\theta))$ is a degree $i-1$ ($\ell-1$) polynomial in $k_j(\theta)$ from some orthogonal family. We then solve for the unknown coefficients $a^i_{j_1\cdots j_n\ell}$. This is a method that can quickly become infeasible because the number of unknown

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

coefficients equals $(n_\theta + 1)(n_k + 1)^n$ for each of the n policy functions. The complete polynomial method uses the form

$$C^i(k, \theta; a) = \sum_{\substack{0 \le j_1 + \cdots + j_n + \ell \le d \\ 0 \le j_i, \ell \le d}} a^i_{j_1 \ldots j_n \ell} \, \varphi_{j_1}(k_1) \cdots \varphi_{j_n}(k_n) \, \psi_\ell(\theta).$$

In this case, the total number of unknown coefficients is a far smaller number; in fact, the number of unknown coefficients here is the same as in the Taylor-series expansion method, those numbers displayed in Table 3. In Table 3, we displayed the general formula for the number of coefficients. Note that it grows polynomially, not exponentially.

These general methods are two that are likely to be of general value. Other methods are not likely to be competitive for smooth models. For example, one would need far more unknown coefficients to use splines.[12]

5.2. Solution: Fixed-Point Iteration vs. Newton's Method vs. Time Iteration

The next critical choice is the method we use to solve the projection equations for the coefficients a. Newton's method[13] treats the conditions $P(a) = 0$ as a system of nonlinear equations and solves for a by repeated quadratic approximations. Newton's method is locally quadratically convergent, but each step uses $\mathcal{O}(n^3)$ time because it computes a Jacobian. Some refinements economize on this by approximating the Jacobian, but the computational cost per step is still a problem.

Fixed-point iteration proceeds more directly, uses fewer computations per step, but has only linear convergence if it converges at all. Specifically, fixed-point iteration takes the policy functions computed in iteration j, $\hat{C}^{i,j}$ and applies the computation

$$\begin{aligned} \hat{C}^{i,j+1}(k, \theta) = {} & (u_i')^{-1}(\beta \hat{E}\{u_i'(\hat{C}^{i,j}(Y(k, \theta) - \hat{C}^{\cdot,j}(k, \theta), \tilde{\theta})) \\ & \times F_1(Y(k, \theta) - \hat{C}^{\cdot,j}(k, \theta))., \tilde{\theta}) \mid \theta\}) \end{aligned} \tag{37}$$

at a finite number of points (k, θ) to produce $\hat{C}^{i,j+1}(k, \theta)$ data sufficient to fix the unknown coefficients of $\hat{C}^{i,j+1}$. Because $\hat{C}^{i,j+1}(k, \theta)$ is expressed directly in terms of the right-hand side of (37), the computation cost is small. Fixed-point iteration can be motivated by learning arguments in Marcet and Sargent (1989), but was actually used in the rational-expectations literature earlier by Miranda and Helmberger (1988) who observed that it was an efficient method for computation.

Time iteration also uses the Euler equation to compute a new value for $\hat{C}^{i,j+1}(k, \theta)$, but instead, the equation

$$\begin{aligned} \hat{C}^{i,j+1}(k, \theta) = {} & (u_i')^{-1}(\beta \hat{E}\{u_i'(\hat{C}^{i,j}(Y(k, \theta) - \hat{C}^{\cdot,j+1}(k, \theta), \tilde{\theta})) \\ & \times F_k((Y(k, \theta) - \hat{C}^{\cdot,j}(k, \theta))., \tilde{\theta}) \mid \theta\}) \end{aligned} \tag{38}$$

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

is used to generate the necessary data. This is a much more complex way to fix $\hat{C}^{i,j+1}(k,\theta)$ values because, for a fixed (k,θ) vector, (38) is a nonlinear equation in $\hat{C}^{i,j+1}(k,\theta)$, hence involving more effort. Both fixed-point iteration and time iteration are only linearly convergent. However, the computational demands of each iteration are only $\mathcal{O}(n^2)$. Time iteration is more reliable but generally slower than fixed-point iteration when the latter converges. Time iteration was used by Gustafson (1958), and by Wright and Williams (1982, 1984). In this paper, we do not discuss time iteration because experience[14] and theory indicate that it will be much slower than Newton's method for small problems and slower than the fixed-point iteration results below.

5.3. Accuracy

We need ways to ascertain if our solutions are "good." To measure the accuracy of our approximations, we evaluate

$$E(a) \equiv \|R^i(.,.,\hat{C}(\cdot;a))/\hat{C}^i(\cdot;a)\|$$

at the solution for a for various norms, and use E as a unit-free measure of the "irrationality" in the approximate solution. For example, if the L_∞ norm is 10^{-2}, then the Euler equation is accurate to within a penny per dollar of expenditure. Another way of expressing this is to say that the approximate policy functions $\hat{C}^i(k,\theta;a)$ comprise an ϵ-equilibrium for $\epsilon = E$. This criterion is a strong one, much stronger than the one used in the Taylor–Uhlig (1990) symposium, which essentially examined $E\{R^i(.,.,\hat{C})\}$ where the expectation is taken over simulated paths for k and θ generated by $\hat{C}$. The norm we compute is nonzero as long as the Euler equation error is nonzero anywhere, and the L_∞ norm is essentially the maximum Euler equation error over the region explored. We parameterize the production function so that the symmetric deterministic steady-state capital stock is 1 for each agent, and then let $k \in [0.5, 1.5]^n$, a large range.

5.4. Results

We now give results for these alternative approaches. In Table 5 we examine 2-, 3-, 4-, and 5-agent models using both tensor product and complete polynomial bases, and using both Newton's method and fixed-point iteration. In all cases, we use product Gaussian quadrature to compute all integrals and we use a simple linear initial guess that goes through the deterministic steady state and zero. The column under γ lists the different values of relative risk aversion for the different agents; these values cover the range generally considered reasonable. We converged when coefficients were deemed within 10^{-5} of the solution. The accuracy is the base 10 logarithm of the L_2 definition of E above applied to the worst-agents Euler equation; in fact, the Euler equation errors were all very close for all agents. The

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

TABLE 5. Time and accuracy comparisons

No. of agents	Tastes (γ)	Degree	Basis	No. of coefficients	Newton's method		Fixed-point iteration	
					Time[a]	Accuracy	Time[a]	Accuracy
1	−2	1	t	4	00:00:05	−2.7	00:00:22	−2.7
			c	3	00:00:06	−2.6	00:00:39	−2.6
		2	t	9	00:00:22	−3.4	00:00:55	−3.4
			c	6	00:00:17	−3.3	00:00:93	−3.3
		3	t	16	00:00:71	−4.1	00:01:15	−4.1
			c	10	00:00:49	−4.0	00:01:92	−4.0
		4	t	25	00:00:70	−4.8	00:02:15	−4.9
			c	30	00:00:99	−4.7	00:03:29	−4.6
2	−1.1	1	t	16	00:00:66	−3.1	00:01:49	−3.1
	−2		c	6	00:00:38	−2.7	00:01:42	−2.7
		2	t	54	00:07:30	−4.1	00:08:02	−4.1
			c	20	00:02:50	−3.4	00:06:43	−3.4
		3	t	128	00:01:22	−5.0	00:32:90	−4.5
			c	40	00:11:40	−4.1	00:20:80	−4.1
		4	t	250	00:12:34	−5.9	00:01:48	−4.5
			c	70	00:45:20	−4.8	00:55:50	−4.7
3	−1.1	1	t	48	00:06:93	−3.4	00:07:50	−3.4
	−2		c	15	00:01:48	−2.8	00:04:60	−2.8
	−3	2	t	243	00:07:07	−4.6	00:02:11	−4.5
			c	63	00:20:80	−3.6	00:36.40	−3.6
		3	t	768	inf	inf	00:19:57	−4.6
			c	105	00:04:05	−4.3	00:03:09	−4.3
		4	t	1875	inf	inf	01:56:00	−4.6
			c	210	00:46:58	−4.9	00:12:45	−4.8
4	−.5	1	t	128	00:01:09	−3.5	00:33:10	−3.5
	−1.1		c	24	00:05:10	−2.9	00:13:30	−2.9
	−2	2	t	972	inf	inf	00:24:57	−4.6
	−3		c	84	00:02:47	−3.7	00:03:04	−3.7
		3	t	4096	inf	inf	07:13:00	−4.6
			c	224	00:52:11	−4.4	00:26:01	−4.4
5	−.5	1	t	320	00:08:52	−3.6	00:02:48	−3.6
	−1.1		c	35	00:17:90	−3.0	00:00:38	−3.0
	−2	2	t	3645	inf	inf	05:16:00	−4.6
	−3		c	140	00:12:18	−3.8	00:10:18	−3.8
	−4	3	t	20,480	inf	inf	inf	inf
			c	420	12:50:00	−4.5	03:27:00	−4.5

[a] 00:00:00 = hours: minutes: seconds; inf = infeasible, t = tensor c = complete.

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

magnitude of the accuracy measure is therefore (roughly speaking) the decimal digit accuracy. These examples were computed on a 50 MHz 486 computer.

The results are as expected. First, the total amount of time for Newton's method is roughly cubic in the number of unknown coefficients. Newton's method is generally more accurate than fixed-point iteration, but for large problems takes much more time. The other problem is that storing the Jacobian can demand more space than available in RAM; this happens for the larger models. Note that the complete polynomial method is generally efficient, but tensor product methods are not.

The fixed-point iteration method turns out to dominate Newton's method for large models. In fact, for the higher-order approximations of the five-agent model, only fixed-point iteration with complete polynomials is feasible, given space requirements. The fixed-point iteration method turns out to be stable (given the initial condition), a fortunate property of these models. Judd (forthcoming) shows that fixed-point iteration applied to the single-agent, deterministic model can be unstable when the elasticity of substitution is large, but is stable near the deterministic steady state for most reasonable choices of γ. This stability property apparently is robust to multiple-agent models.

6. HETEROGENEOUS WEALTH, COMMON TASTES

A simpler case is where all agents have the same intertemporal tastes, but may have different wealth. Here, we can exploit obvious symmetry properties to drastically reduce the computational burden. We now assume that the n different agents have the same utility function, $u(c)$, and a common discount factor β. Outside of some special cases, we cannot aggregate preferences to reduce the model to a representative agent. Let $C^i(k)$ be the consumption of type i agents when the wealth distribution is $k = (k_1, k_2, \ldots, k_n)$, where k_i is the capital owned by a type-i agent. The equilibrium is again defined by the collection of Euler equations

$$u_i'(C^i(k,\theta)) = \beta E\{u_i'(C^i(Y(k,\theta) - C(k,\theta), \tilde{\theta}))\, F_1((Y(k,\theta) - C(k,\theta))., \tilde{\theta}) \mid \theta\} \tag{39}$$

for $i = 1, 2, \ldots, n$, where $Y(k,\theta) \in R^n$ is the distribution of income in a period with initial capital stock k and productivity θ, that is, $Y^i(k,\theta) = k_i F_1(k_\cdot, \theta) + w(k,\theta)$. The tensor method approximates each consumption function as

$$C^\ell(k,\theta;a) = \sum_{i_1=0}^{n_k} \cdots \sum_{i_n=0}^{n_k} \sum_{j=0}^{n_\theta} a^\ell_{i_1\cdots i_n j} \varphi_{i_1}(k_1) \cdots \varphi_{i_n}(k_n) \psi_j(\theta),$$

where $\varphi_i(k_m)$ ($\psi_j(\theta)$) is a degree $i-1$ ($\ell - 1$) polynomial in k_m (θ) from some orthogonal family. We then solve for the unknown coefficients $a^\ell_{i_1\cdots i_n j}$. However, we can impose certain symmetry conditions. First, an agent's behavior depends only on his wealth and the distribution of wealth, and all agents have the same consumption function. Therefore, if $i_\ell = i_{\ell'}$, then the coefficients $a^\ell_{i_1\cdots i_\ell\cdots i_n j} = a^{\ell'}_{i_1\cdots i_{\ell'}\cdots i_n j}$ for all

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

TABLE 6. Symmetric polynomials

Degree	Polynomials
1	$x + y + \cdots + z$
2	$x^2 + y^2 + \cdots + z^2,\ (x + y + \cdots + z)^2$
3	$x^3 + y^3 + \cdots + z^3,\ x^2y + x^2z + \cdots + y^2z + \cdots,\ (x + y + \cdots + z)^3$

ℓ and ℓ'. Furthermore, $a^1_{i_1\cdots i_\ell\cdots i_n j} = a^1_{i_1\pi(i_2)\cdots\pi(i_\ell)\cdots\pi(i_n)j}$ for all permutations π on $\{i_2, i_3, \ldots, i_n\}$. Together, these conditions imply that the policy function for type 1 agents is of the form

$$C^1(k, \theta; a) = \sum_{i,j,\ell} a_{ij\ell}\varphi_i(k_1)\Psi_j(k_2, k_3, \ldots, k_n)\psi_\ell(\theta),$$

where $\Psi_j(k_2, k_3, \ldots, k_n)$ is a symmetric polynomial in the $n - 1$ variables k_2, $k_3, \ldots, k_n$. The degree 1, 2, and 3 symmetric polynomials are listed in Table 6.

The complete polynomial method uses, for a type-1 agent, the form

$$C^1(k, \theta; a) = \sum_{0\le\deg(\phi_i)+\deg(\Psi_j)+\deg(\psi_\ell)\le n} a_{ij\ell}\varphi_i(k_1)\Psi_j(k_2, k_3, \ldots, k_n)\,\psi_\ell(\theta),$$

where the consumption functions of the other types are constructed by symmetry considerations. To identify the unknown coefficients, we use the identifying projections

$$0 = \int\cdots\int u'(C^1(k,\theta)) - \beta E\,\{u'(C^1(Y(k,\theta) - C(k,\theta), \tilde\theta))$$
$$\times F_1((Y(k,\theta) - C(k,\theta)., \tilde\theta) \mid \theta\}\,\varphi_i(k_1)\Psi_j(k_2, k_3, \ldots, k_n)\psi_\ell(\theta)\,dk\,d\theta$$

for i, j, and ℓ such that $0 \le \deg(\phi_i) + \deg(\Psi_j) + \deg(\psi_\ell) \le n$. In this case, the total number of unknown coefficients is a far smaller number than the unrestricted case, and there is only one Euler equation that needs to be fitted. The result is a far smaller system than we would have if we had followed the general multi-agent approach of the preceding section. To save on space, we do not discuss computational results because they are exactly what one would expect: imposing the symmetry conditions results in far faster computations without any loss of accuracy.

7. CONCLUSION

This paper has shown that it is feasible to compute rational-expectations models substantially more complex than the usual representative-agent, single-good model. Using only 486- and Pentium-class personal computers, we have shown how to use perturbation methods to solve dynamic models with up to 50 state variables (that is, 50 agents or 50 capital stocks, or some combination) in only

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

TABLE 7. Final comparisons

Method	Basis	Solution method	Advantages	Disadvantages
Taylor series	Complete	Eigenvalues, linear equations	Very fast	Local validity
Projection methods	Tensor or complete	Newton	Quadratic convergence	Infeasible for large problems
	Tensor or complete	Successive approximation	Easy iterations	Possible nonconvergence

minutes. We also have demonstrated that more globally oriented methods can produce accurate approximations of multiple-agent models.

We also have seen that the methods of choice depend on the size of the problem. Table 7 summarizes our findings. Our experiments indicate that the dominant approach to large models with Euler equation formulations will be fixed-point iteration solution methods combined with complete polynomial bases, outperforming Newton-based methods and monotone- and contraction-operator methods. The apparent dominance of fixed-point iteration indicates that we need to better understand the stability problems of fixed-point iteration and develop approaches to detect and deal with them.

This work clearly indicates that rational expectations models of moderate size can be reliably and quickly solved numerically. This paper focused on the approximation and solution methods that have been used, and used only the simplest integration methods. Exploitation of more advanced approximation, solution, and integration techniques surely will lead to drastic improvements. We also suspect that these methods are capable of making efficient use of supercomputing environments, making even larger models feasible. Overall, given the limited use of available hardware and software in this paper, we believe that numerical solution of large stochastic, dynamic models is an attainable goal.

NOTES

1. The term projection method is a catchall term in the mathematical literature that includes method of weighted residuals, finite element, boundary element, Galerkin, least squares, Rayleigh–Ritz, and other similar methods.

2. Wright and Williams (1984) and some later writers parameterize a conditional-expectation function, such as the one on the right-hand side of (2), that characterizes the solution. If one is to use polynomial approximation methods, then one should approximate some continuous function that characterizes equilibrium. Wright and Williams show that when price and policy functions have kinks, it is better to approximate the conditional-expectations function. Our comments apply equally to this case because, as is clear from Judd (1992), the key fact is that one is solving for an unknown smooth function.

3. For example, the procedure we outline is *not* the procedure discussed by McGrattan (1990); in general, our Taylor-series expansion method produces different results.

https://doi.org/10.1017/S1365100597002022 Published online by Cambridge University Press

4. By "correct," we refer to the concepts, definitions, methods, and solutions derived in the mathematical literature on the stable manifold theorem and asymptotics. The interested reader should consult Coddington and Levinson (1965), Fleming (1971), Fleming and Souganides (1986), and Bensoussan (1988). Judd (forthcoming) gives an example of where one proposed method goes wrong, and a more extensive discussion of these points.

5. Marcet (1994) did not indicate what he meant by perturbation methods. It is true that some of the linear approximation methods discussed in the macroeconomic literature are not extended easily to compute higher-order terms. However, it has always been known in the mathematical literature that higher-order terms can be computed easily if one used what the mathematical literature calls perturbation methods; see Bensoussan (1988) and the extensive citations there, for example.

6. This is also the method used in dynamic analyses of the 1970's and early 1980's, such as Hall (1971), Fischer (1979), Judd (1982), Laitner (1984), and many others. Many later writers reject the standard approach in favor of ad hoc procedures, but never explain why. For example, Cooley and Hansen's (1989) analysis of monetary equilibria near the deterministic steady state ignores the simpler procedure outlined by Fischer (1979).

7. Taylor-series expansions are valid over nontrivial intervals for analytic functions, but we make no claim that our problems have analytic solutions.

8. By a "linear theory of u," we mean an approximation of u that is linear in the state variables x, and that is asymptotically valid as x converges to the steady-state value of x.

9. The discrete-time approach can be analyzed similarly, but at a greater notational cost.

10. This estimate is based on log interpolation of the $n = 10$ and $n = 100$ cases in the Eigenvector results in Table 1.

11. Again, this is based on log interpolating in the Linear system column in Table 1 between the $n = 10$ and $n = 100$ cases.

12. One possible alternative is to create a problem-specific basis, an approach defined and discussed by Judd (1996) and papers cited therein.

13. We actually use an implementation of Powell's hybrid method. Our experience is that Powell's method is far faster than the pure Newton method. We have tried a few other methods on various projection problems but only with disappointing results.

14. For example, compare the execution times in Coleman's (1990) implementation of time iteration and the Newton method results reported by Judd (1992). The experience with time iteration in Bizer and Judd (1989) also was disappointing.

REFERENCES

Bensoussan, A. (1988) *Perturbation Methods in Optimal Control*. New York: John Wiley & Sons.

Bizer, D.S., & K.L. Judd (1989) Uncertainty and taxation. *American Economic Review* 79, 331–336.

Blanchard, O.J. & C.M. Kahn (1980) The solution of linear difference models under rational expectations. *Econometrica* 48, 1305–1311.

Brock, W.A. & L.J. Mirman (1972) Optimal economic growth and uncertainty: The discounted case. *Journal of Economic Theory* 4, 479–513.

Brock, W.A. & S.J. Turnovsky (1981) The analysis of macroeconomic policies in perfect foresight equilibrium. *International Economic Review* 22, 179–209.

Christiano, L.J. (1990) Solving the stochastic growth model by linear-quadratic approximation and by value-function iteration. *Journal of Business and Economic Statistics* 8, 23–26.

Coddington, E.A. & N. Levinson (1965) *Theory of Ordinary Differential Equations*. New York: McGraw–Hill.

Coleman, W.J., II (1990) Solving the stochastic growth model by policy function iteration. *Journal of Business and Economic Statistics* 8, 27–29.

den Haan, W. & A. Marcet (1990) Solving the stochastic growth model by parameterizing expectations. *Journal of Business and Economic Statistics* 8, 31–34.

Fischer, S. (1979) Capital accumulation on the transition path in a monetary optimizing model. *Econometrica* 47, 1433–1439.

Fleming, W. (1971) Stochastic control for small noise intensities. *SIAM Journal of Control* 9, 473–517.
Fleming, W. & P.E. Souganides (1986) Asymptotic series and the method of vanishing viscosity. *Indiana University Mathematics Journal* 35, 425–447.
Gustafson, R.L. (1958) Carryover Levels for Grains: A Method for Determining Amounts That Are Optimal Under Specified Conditions. USDA Technical Bulletin 1178.
Hall, R. (1971) The dynamic effects of fiscal policy in an economy with foresight. *Review of Economic Studies* 38, 229–244.
Judd, K.L. (1982) An alternative to steady-state comparisons in perfect foresight models. *Economics Letters* 10, 55–59.
Judd, K.L. (1992) Projection methods for solving aggregate growth models. *Journal of Economic Theory* 58, 410–452.
Judd, K.L. (1996) Approximation, perturbation, and projection methods in economic analysis. In H. Amman, D. Kendrick, & J. Rust (eds.), *Handbook of Computational Economics*. North–Holland.
Judd, K.L. (forthcoming) *Numerical Methods in Economics*. MIT Press.
Judd, K.L. & S.-M. Guu (1993) Perturbation solution methods for economic growth models. In Hal Varian (ed.), *Economic and Financial Modeling with Mathematica*. New York: Springer–Verlag.
Judd, K.L. & S.-M. Guu (forthcoming) Asymptotic methods for aggregate growth models. *Journal of Economic Dynamics and Control*.
Kydland, F.E. & E.C. Prescott (1982) Time to build and aggregate fluctuations. *Econometrica* 50, 1345–1370.
Laitner, J. (1984) Transition time paths for overlapping-generations models. *Journal of Economic Dynamics and Control* 7, 111–129.
Magill, J.P.M. (1977) A local analysis of N-sector capital accumulation under uncertainty. *Journal of Economic Theory* 15, 211–219.
Marcet, A. (1994) Simulation analysis of dynamic stochastic models: Applications to theory and estimation. In C. Sims (ed.), *Advances in Econometrics Sixth World Congress*, Vol. II, pp. 91–118, Cambridge, U.K.: Cambridge University Press.
Marcet, A. & T.J. Sargent (1989) Convergence of least squares learning mechanisms in self-referential linear stochastic models. *Journal of Economic Theory* 48, 337–368.
McGrattan, E.R. (1990) Solving the stochastic growth model by linear-quadratic approximation. *Journal of Business and Economic Statistics* 8, 41–43.
Miranda, M.J. & P.G. Helmberger (1988) The effects of commodity price stabilization programs. *American Economic Review* 78, 46–58.
Stokey, N.L. & R.E. Lucas, Jr. (1989) *Recursive Methods in Economic Dynamics*. Cambridge, MA: Harvard University Press.
Taylor, J.B. & H. Uhlig (1990) Solving nonlinear stochastic growth models: A comparison of alternative solution methods. *Journal of Business and Economic Statistics* 8, 1–18.
Wright, B.D. & J.C. Williams (1982) The economic role of commodity storage. *Economic Journal* 92, 596–614.
Wright, B.D. & J.C. Williams (1984) The welfare effects of the introduction of storage. *Quarterly Journal of Economics* 99, 169–182.
Williams, J. & B. Wright (1991) *Storage and Commodity Markets* Cambridge, U.K.: Cambridge University Press.