Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER Journal of Economic Dynamics & Control 30 (2006) 2477–2508
JOURNAL OF Economic Dynamics & Control
www.elsevier.com/locate/jedc

Comparing solution methods for dynamic equilibrium economies

S. Borağan Aruoba[a], Jesús Fernández-Villaverde[b,*], Juan F. Rubio-Ramírez[c]

[a]*University of Maryland, USA*
[b]*Department of Economics, University of Pennsylvania, 160 McNeil Building, 3718 Locust Walk, Philadelphia, PA 19104, USA*
[c]*Federal Reserve Bank of Atlanta, USA*

Received 23 November 2003; accepted 26 July 2005
Available online 1 December 2005

Abstract

This paper compares solution methods for dynamic equilibrium economies. We compute and simulate the stochastic neoclassical growth model with leisure choice using first, second, and fifth order perturbations in levels and in logs, the finite elements method, Chebyshev polynomials, and value function iteration for several calibrations. We document the performance of the methods in terms of computing time, implementation complexity, and accuracy, and we present some conclusions based on the reported evidence.

JEL classification: C63; C68; E37

Keywords: Dynamic equilibrium economies; Computational methods; Linear and nonlinear solution methods

*Corresponding author. Tel.: +1 219 898 15 04; fax: +1 215 573 20 57.
E-mail address: jesusfv@econ.upenn.edu (J. Fernández-Villaverde).

0165-1889/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.jedc.2005.07.008

1. Introduction

This paper addresses the following question: how different are the computational answers provided by alternative solution methods for dynamic equilibrium economies?

Most dynamic models do not have an analytic, closed-form solution, and we need to use numerical methods to approximate their behavior. There are a number of procedures for undertaking this task (see Judd, 1998; Marimón and Scott, 1999; or Miranda and Fackler, 2002). However, it is difficult to assess a priori how the quantitative characteristics of the computed equilibrium paths change when we move from one solution approach to another. Also, the relative accuracies of the approximated equilibria are not well understood.

The properties of a solution method are not only of theoretical interest but crucial to assessing the reliability of the answers provided by quantitative exercises. For example, if we state, as in the classical measurement by Kydland and Prescott (1982), that productivity shocks account for 70 percent of the fluctuations in the U.S. economy, we want to know that this number is not a by-product of numerical error. Similarly, if we want to estimate the model, we need an approximation that does not bias the estimates, but yet is quick enough to make the exercise feasible.

Over 15 years ago a group of researchers compared solution methods for the growth model without leisure choice (see Taylor and Uhlig, 1990 and the companion papers). Since then, a number of non-linear solution methods — several versions of projection (Judd, 1992) and perturbation procedures (Judd and Guu, 1997) — have been proposed as alternatives to more traditional (and relatively simpler) linear approaches and to value function iteration. However, little is known about the relative performance of the new methods.[1] This is unfortunate since these new methods, built on the long experience of applied mathematics, promise superior performance. This paper tries to fill part of this gap in the literature.

To do so, we use the canonical stochastic neoclassical growth model with leisure choice. We understand that our findings are conditional on this concrete choice and that this paper cannot substitute for the close examination that each particular problem deserves. The hope is that, at least partially, the lessons from our application could be useful for other models. In that sense we follow a tradition in numerical analysis that emphasizes the usefulness of comparing the performance of algorithms in well-known *test problems*.

Why do we choose the neoclassical growth model as our test problem? First, because it is the workhorse of modern macroeconomics. Any lesson learned in this context is bound to be useful in a large class of applications. Second, because it is simple, a fact that allows us to solve it with a wide range of methods. For example, a

[1]For the growth model we are only aware of the comparison between Chebyshev polynomials and different versions of the dynamic programming algorithm and policy iteration undertaken by Santos (1999) and Benítez-Silva et al. (2000). However, the two papers (except one case in Santos, 1999) deal with the model with full depreciation and never with other nonlinear methods. In a related contribution, Christiano and Fisher (2000) evaluate how projection methods deal with models with occasionally binding constraints.

model with binding constraints would rule out perturbation methods. Third, because we know a lot about the theoretical properties of the model, results that are useful for interpreting our findings. Finally, because there exists a current project organized by Den Haan, Judd, and Julliard to compare different solution methods in heterogeneous agents economies. We see our paper as a complement to this project.

We solve and simulate the model using two main approaches: perturbation and projection algorithms. Within perturbation, we consider first, second, and fifth order, both in levels and in logs. Note that a first order perturbation is equivalent to linearization when performed in levels and to loglinearization when performed in logs. Within projection we consider finite elements and Chebyshev polynomials. For comparison purposes, we also solve the model using value function iteration. This last choice is a natural benchmark given our knowledge about the convergence properties of value function iteration (Santos and Vigo, 1998).

We report results for a benchmark calibration of the model and for alternative calibrations that change the variance of the productivity shock and the risk aversion. In that way we study the performance of the methods both for a nearly linear case (the benchmark calibration) and for highly nonlinear cases (high variance/high risk aversion). In our simulations we keep a fixed set of shocks common for all methods. That allows us to observe the dynamic responses of the economy to the same driving process and how computed paths and their moments differ for each approximation. We also assess the accuracy of the solution methods by reporting Euler equation errors in the spirit of Judd (1992).

Five main results deserve to be highlighted. First, perturbation methods deliver an interesting compromise between accuracy, speed, and programming burden. For example, we show how a fifth order perturbation has an advantage in terms of accuracy over all other solution methods for the benchmark calibration. We quantitatively assess how much and how quickly perturbations deteriorate when we move away from the steady state (remember that perturbation is a local method). Also, we illustrate how the simulations display a tendency to explode and the reasons for such behavior.

Second, since higher order perturbations display a much superior performance over linear methods for a trivial marginal cost, we see a compelling reason to move some computations currently undertaken with linear methods to at least a second order approximation.

Third, even if the performance of linear methods is disappointing along a number of dimensions, linearization in levels is preferred to loglinearization for both the benchmark calibration and the highly nonlinear cases. The result contradicts a common practice based on the fact that the exact solution to the model with log utility, inelastic labor, and full depreciation is loglinear.

Fourth, finite elements perform very well for all parameterizations. It is extremely stable and accurate over the range of the state space even for high values of the risk aversion and the variance of the shock. This property is crucial in estimation procedures where accuracy is required to obtain unbiased estimates (see Fernández-Villaverde and Rubio-Ramírez, 2004). Also, we use simple linear basis functions. Given the smoothness of the solution, finite elements with higher order basis

functions would do even better. However, finite elements suffer from being probably the most complicated to implement in practice (although not the most intensive in computing time).

Fifth, Chebyshev polynomials share all the good results of the finite elements method and are easier to implement. Since the neoclassical growth model has smooth policy functions, it is not surprising that Chebyshev polynomials do well in this application. However in a model where policy functions has complicated local behavior, finite elements might outperform Chebyshev polynomials.

Therefore, although our results depend on the particular model we have used, they should encourage a wider use of perturbation, to suggest the reliance on finite elements for problems that demand high accuracy and stability, and support the progressive phasing out of pure linearizations.

The rest of the paper is organized as follows. Section 2 presents the neoclassical growth model. Section 3 describes the different solution methods used to approximate the policy functions of the model. Section 4 discusses the benchmark calibration and alternative robustness calibrations. Section 5 reports numerical results and Section 6 concludes.

2. The stochastic neoclassical growth model

We use the basic model in macroeconomics, the stochastic neoclassical growth model with leisure, as our test case for comparing solution methods.[2]

Since the model is well known, we go through only the exposition required to fix notation. There is a representative household with utility function from consumption, c_t, and leisure, $1-l_t$:

$$U = \mathrm{E}_0 \sum_{t=1}^{\infty} \beta^{t-1} \frac{\left(c_t^{\theta}(1-l_t)^{1-\theta}\right)^{1-\tau}}{1-\tau},$$

where $\beta \in (0,1)$ is the discount factor, τ is the elasticity of intertemporal substitution, θ controls labor supply, and E_0 is the conditional expectation operator. The model requires this utility function to generate a balanced growth path with constant labor supply, as we observe in the post-war U.S. data. Also, this function nests a log utility as $\tau \to 1$.

There is one good in the economy, produced according to $y_t = \mathrm{e}^{z_t} k_t^{\alpha} l_t^{1-\alpha}$, where k_t is the aggregate capital stock, l_t is aggregate labor, and z_t is a stochastic process representing random technological progress. The technology follows the process $z_t = \rho z_{t-1} + \varepsilon_t$ with $|\rho| < 1$ and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. Capital evolves according to the law of motion $k_{t+1} = (1-\delta)k_t + i_t$, where δ is the depreciation rate and i_t investment. The economy must satisfy the aggregate resource constraint $y_t = c_t + i_t$.

[2] An alternative could have been the growth model with log utility and full depreciation, a case where a closed-form solution exists. However, it would be difficult to extrapolate the lessons from this example into statements for the general case. Santos (2000) shows how changes in the curvature of the utility function and depreciation quite influence the size of the Euler equation errors.

Both welfare theorems hold in this economy. Consequently, we can solve directly for the social planner's problem where we maximize the utility of the household subject to the production function, the evolution of technology, the law of motion for capital, the resource constraint, and some initial k_0 and z_0.

The solution to this problem is fully characterized by the equilibrium conditions:

$$\frac{\left(c_t^{\theta}(1-l_t)^{1-\theta}\right)^{1-\tau}}{c_t} = \beta \mathrm{E}_t\left\{\frac{\left(c_{t+1}^{\theta}(1-l_{t+1})^{1-\theta}\right)^{1-\tau}}{c_{t+1}}\left(1+\alpha \mathrm{e}^{z_{t+1}}k_{t+1}^{\alpha-1}l_{t+1}^{1-\alpha}-\delta\right)\right\}, \quad (1)$$

$$(1-\theta)\frac{\left(c_t^{\theta}(1-l_t)^{1-\theta}\right)^{1-\tau}}{1-l_t} = \theta\frac{\left(c_t^{\theta}(1-l_t)^{1-\theta}\right)^{1-\tau}}{c_t}(1-\alpha)\mathrm{e}^{z_t}k_t^{\alpha}l_t^{-\alpha}, \quad (2)$$

$$c_t + k_{t+1} = \mathrm{e}^{z_t}k_t^{\alpha}l_t^{1-\alpha} + (1-\delta)k_t, \quad (3)$$

$$z_t = \rho z_{t-1} + \varepsilon_t. \quad (4)$$

The first equation is the standard Euler equation that relates current and future marginal utilities from consumption. The second equation is the static first order condition between labor and consumption. The last two equations are the resource constraint of the economy and the law of motion of technology.

Solving for the equilibrium of this economy amounts to finding three policy functions for consumption $c(\cdot,\cdot)$, labor $l(\cdot,\cdot)$, and next period's capital $k'(\cdot,\cdot)$ that deliver the optimal choice of the variables as functions of the two state variables, capital and technology.

All the solution methods described in the next section, except value function iteration, exploit the equilibrium conditions (1)–(4) to find these functions. This characteristic makes the extension of the methods to non-Pareto optimal economies — where we need to solve directly for the market allocation — straightforward. Thus, we can export at least part of the intuition from our results to a large class of economies.

Also, from Eqs. (1)–(4), we compute the model's steady state: $k_{ss} = \Psi/(\Omega+\varphi\Psi)$, $l_{ss} = \varphi k_{ss}$, $c_{ss} = \Omega k_{ss}$, and $y_{ss} = k_{ss}^{\alpha}l_{ss}^{1-\alpha}$, where $\varphi = (1/\alpha(1/\beta-1+\delta))^{1/(1-\alpha)}$, $\Omega = \varphi^{1-\alpha}-\delta$, and $\Psi = \theta/(1-\theta)(1-\alpha)\varphi^{-\alpha}$. These values will be useful below.

3. Solution methods

The system of equations listed above does not have a known analytical solution. We need to employ a numerical method to solve it.

The most direct approach to do so is to attack the social planner's problem with value function iteration. This procedure is safe, reliable, and enjoys useful convergence properties (Santos and Vigo, 1998). However, it is extremely slow (see Rust, 1996, 1997 for accelerating algorithms) and suffers from a strong curse of the dimensionality. Also, it is difficult to adapt to non-Pareto optimal economies (see Kydland, 1989).

Because of these problems, the development of new solution methods for dynamic models has become an important area of research during the last decades. Most of these procedures can be grouped into two main approaches: perturbation and projection algorithms.

Perturbation methods build a Taylor series expansion of the agents' policy functions around the steady state of the economy and a perturbation parameter. In two seminal papers, Hall (1971) and Magill (1977) showed how to compute the first term of this series. Since the policy resulting from a first order approximation is linear and many dynamic models display behavior that is close to a linear law of motion, the approach became quite popular under the name of linearization. Judd and Guu (1993) extended the method to compute the higher-order terms of the expansion.

The second approach is projection methods (Judd, 1992; Miranda and Helmberger, 1988). These methods take basis functions to build an approximated policy function that minimizes a residual function (and, hence, are also known as minimum weighted residual methods). There are two versions of the projection methods. In the first one, called finite elements, the basis functions are nonzero only locally. In the second, called spectral, the basis functions are nonzero globally.

Projection and perturbation methods are attractive because they are much faster than value function iteration while maintaining good convergence properties. This point is of practical relevance. For instance, in estimation problems, speed is of the essence since we may need to repeatedly solve the policy function of the model for many different parameter values. Convergence properties assure us that, up to some accuracy level, we are indeed getting the correct equilibrium path for the economy.

In this paper we compare eight different methods. Using perturbation, we compute a first, second, and fifth order expansion of the policy function in levels. We also compute a first and a second order expansion of the policy function in logs. Using projection, we compute a finite elements method with linear functions and a spectral procedure with Chebyshev polynomials. Finally, and for comparison purposes, we perform a value function iteration.

We do not try to cover every single known method but rather to be selective and choose those methods that we find more promising based either on the literature or on intuition from numerical analysis. Below we discuss how several apparently excluded methods are particular cases of some of our approaches.

The rest of this section describes each of these solution methods. A companion web page at `http://www.econ.upenn.edu/~jesusfv/companion.htm` posts online all the codes required to reproduce the computations, as well as some additional material.

3.1. Perturbation

Perturbation methods (Judd and Guu, 1993; Gaspar and Judd, 1997) build a Taylor series expansion of the policy functions of the agents around the steady state of the economy and a perturbation parameter. In our application we use the standard deviation of the innovation to the productivity level, σ, as the perturbation

parameter. As shown by Judd and Guu (2001), the standard deviation needs to be the perturbation parameter in discrete time models, since odd moments may be important.

Thus, the policy functions for consumption, labor, and capital accumulation are:

$$c_p(k,z,\sigma) = \sum_{i,j,m} a^c_{ijm}(k-k_{ss})^i(z-z_{ss})^j\sigma^m,$$

$$l_p(k,z,\sigma) = \sum_{i,j,m} a^l_{ijm}(k-k_{ss})^i(z-z_{ss})^j\sigma^m,$$

$$k'_p(k,z,\sigma) = \sum_{i,j,m} a^k_{ijm}(k-k_{ss})^i(z-z_{ss})^j\sigma^m,$$

where

$$a^c_{ijm} = \left.\frac{\partial^{i+j+m}c(k,z,\sigma)}{\partial k^i\partial z^j\partial\sigma^m}\right|_{k_{ss},z_{ss},0}, \quad a^l_{ijm} = \left.\frac{\partial^{i+j+m}l(k,z,\sigma)}{\partial k^i\partial z^j\partial\sigma^m}\right|_{k_{ss},z_{ss},0} \quad \text{and}$$

$$a^k_{ijm} = \left.\frac{\partial^{i+j+m}k'(k,z,\sigma)}{\partial k^i\partial z^j\partial\sigma^m}\right|_{k_{ss},z_{ss},0}$$

are equal to the derivative of the policy functions evaluated at the steady state and $\sigma = 0$.

The perturbation scheme works as follows. We take the model equilibrium (1)–(4) and substitute the unknown policy functions $c_p(k,z,\sigma)$, $l_p(k,z,\sigma)$, and $k'_p(k,z,\sigma)$ into them. Then, we take successive derivatives with respect to the k, z, and σ. Since the equilibrium conditions are equal to zero for any value of k, z, and σ, a system created by their derivatives of any order will also be equal to zero. Evaluating the derivatives at the steady state and $\sigma = 0$ delivers a system of equations on the unknown coefficients a^c_{ijm}, a^l_{ijm}, and a^k_{ijm}.

The solution of these systems is simplified because of the recursive structure of the problem. The constant terms a^c_{000}, a^l_{000}, and a^k_{000} are equal to the steady state for consumption, labor, and capital. Substituting these terms in the system of first derivatives of the equilibrium conditions generates a quadratic matrix-equation on the first order terms of the policy function (by nth order terms of the policy function we mean a^q_{ijm} such that $i+j+m=n$ for $q=c,l,k$). Out of the two solutions we pick the one that gives us the stable path of the model.

The next step is to plug the coefficients found in the previous two steps in the system created by the second order expansion of the equilibrium conditions. This generates a linear system in the second order terms of the policy function that is trivial to solve.

Iterating in the procedure (taking a one higher order derivative, substituting previously found coefficients, and solving for the new unknown coefficients), we would see that all the higher than second order coefficients are the solution to linear systems. The intuition for why only the system of first derivatives is quadratic is as follows. The neoclassical growth model has two saddle paths. Once we have picked

the right path with the stable solution in the first order approximation, all the other terms are just refinements of this path.

Perturbations deliver an asymptotically correct expression around the deterministic steady state for the policy function. However, the positive experience of asymptotic approximations in other fields of applied mathematics suggests there is the potential for good nonlocal behavior (Bender and Orszag, 1999).

The burden of the method is taking all the required derivatives, since paper and pencil become virtually infeasible after the second derivatives. Gaspar and Judd (1997) show that higher order numerical derivatives accumulate enough errors to prevent their use. An alternative is to work with symbolic manipulation software such as *Mathematica*,[3] as we do, or with specially developed code as the package *PertSolv* written by Jin (Judd and Jin, 2004).

We have to make two decision when implementing perturbation. First, we need to decide the order of the perturbation, and, second, we need to choose whether to undertake our perturbation in levels and logs (i.e., substituting each variable x_t by $x_{ss}e^{\widehat{x}_t}$, where $\widehat{x}_t = \log x_t/x_{ss}$, and obtain an expansion in terms of $\widehat{x}_t$ instead of x_t).

About the first of the issues, we choose first, second, and fifth order perturbations. First order perturbations are exactly equivalent to linearization, probably the most extended procedure to solve dynamic models.[4] Linearization delivers a linear law of motion for the choice variables that displays certainty equivalence, i.e., it does not depend on σ. This point will be important when we discuss our results. Second order approximations have received attention because of the easiness of their computation (Sims, 2000). We find it of interest to assess how much we gain by this simple correction of the linear policy functions. Finally, we pick a high order approximation. After the fifth order the coefficients are nearly equal to the machine zero (in a 32-bit architecture of standard PCs) and further terms do not add much to the approximation.

Regarding the level versus logs choice, some practitioners have favored logs because the exact solution of the neoclassical growth model in the case of log utility and full depreciation is loglinear. Evidence in Christiano (1990) and Den Haan and Marcet (1994) suggests that this may be the right practice but the question is not completely settled. To cast light on this question, we computed our perturbations both in levels and in logs.

Because of space considerations, we present results only in levels except for two cases: the first order approximation in logs (also known as loglinerization) because it is commonly employed, and the second order approximation for a high variance/high risk aversion case, because in this parametrization the results depend on the use

[3]For second order perturbations we can also use the *Matlab* programs by Schmitt-Grohé and Uribe (2004) and Sims (2000). For higher order perturbations we used *Mathematica* because the symbolic toolbox of *Matlab* cannot handle more than the second derivatives of abstract functions.

[4]Note that, subject to applicability, all different linear methods-linear quadratic approximation (Kydland and Prescott, 1982), the eigenvalue decomposition (Blanchard and Kahn, 1980; King et al., 2002), generalized Schur decomposition (Klein, 2000), or the QZ decomposition (Sims, 2002) among many others, deliver exactly the same result as the first order perturbation. The linear approximation of a differentiable function is unique and invariant to differentiable parameters transformations.

of levels or logs. In the omitted cases, the results in logs were nearly indistinguishable from the results in levels.

3.2. Projection methods

Now we present two different versions of the projection algorithm: the finite elements method and the spectral method with Chebyshev polynomials.

3.2.1. Finite elements method

The finite elements method (Hughes, 2000) is the most widely used general-purpose technique for numerical analysis in engineering and applied mathematics. The method searches for a policy function for labor supply of the form $l_{fe}(k,z;\overline{\theta}) = \sum_{i,j}\overline{\theta}_{ij}\Psi_{ij}(k,z)$ where $\Psi_{ij}(k,z)$ is a set of basis functions and $\overline{\theta}$ is a vector of parameters to be determined. Given $l_{fe}(k,z;\overline{\theta})$, the static first order condition, (2), and the resource constraint, (3), imply two policy functions $c(k,z;l_{fe}(k,z;\overline{\theta}))$ and $k'(k,z;l_{fe}(k,z;\overline{\theta}))$ for consumption and next period capital.

The essence is to select basis functions that are zero for most of the state space except a small part of it, known as 'element', an interval in which they take a simple form, typically linear.[5] Beyond being conceptually intuitive, this choice of basis functions features several interesting properties. First, it provides a lot of flexibility in the grid generation: we can create smaller elements (and consequently very accurate approximations of the policy function) where the economy spends more time and larger ones in those areas less travelled. Second, since the basis functions are nonzero only locally, large numbers of elements can be handled. Third, the finite elements method is well suited for implementation in parallel machines.

The implementation of the method begins by writing the Euler equation as:

$$U_{c,t} = \frac{\beta}{\sqrt{2\pi\sigma}}\int_{-\infty}^{\infty}\left[U_{c,t+1}\left(1+\alpha e^{z_{t+1}}k_{t+1}^{\alpha-1}l_{fe}(k_{t+1},z_{t+1})^{1-\alpha}-\delta\right)\right] \times \exp\left(-\frac{\varepsilon_{t+1}^2}{2\sigma^2}\right)\mathrm{d}\varepsilon_{t+1}, \tag{5}$$

where

$$U_{c,t} = \frac{\left(c(k_t,z_t;l_{fe}(k_t,z_t;\overline{\theta}))^{\theta}(1-l_{fe}(k_t,z_t;\overline{\theta}))^{1-\theta}\right)^{1-\tau}}{c(k_t,z_t;l_{fe}(k_t,z_t;\overline{\theta}))},$$

$k_{t+1} = k'(k_t,z_t;l_{fe}(k_t,z_t;\overline{\theta}))$, and $z_{t+1} = \rho z_t + \varepsilon_{t+1}$.

We use the Gauss–Hermite method (Press et al., 1992) to compute the integral of the right-hand side of Eq. (5). Hence, we need to bound the domain of the state variables. To bound the productivity level of the economy define $\lambda_t = \tanh(z_t)$. Since $\lambda_t \in [-1,1]$, we have $\lambda_t = \tanh(\rho\tanh^{-1}(\lambda_{t-1}) + \sqrt{2}\sigma v_t)$, where $v_t = \varepsilon_t/\sqrt{2}\sigma$. Now,

[5]We could use higher order basis functions. However, these schemes, known as the *p-method*, are much less used than the so-called *h-method,* whereby the approximation error is reduced by specifying smaller elements.

since

$$\exp(z_{t+1}) = \frac{\sqrt{1+\lambda_{t+1}}}{\sqrt{1-\lambda_{t+1}}} = \widehat{\lambda}_{t+1},$$

we rewrite (5) as

$$U_{c,t} = \frac{\beta}{\sqrt{\pi}} \int_{-1}^{1} \left[U_{c,t+1}(1 + \alpha \widehat{\lambda}_{t+1} k_{t+1}^{\alpha-1} l_{fe}(k_{t+1}, \tanh^{-1}(\lambda_{t+1}))^{1-\alpha} + \delta) \right] \times \exp(-v_{t+1}^2)\, \mathrm{d}v_{t+1}, \tag{6}$$

where

$$U_{c,t} = \frac{\left(c(k_t, \tanh^{-1}(\lambda_t); l_{fe}(k_t, \tanh^{-1}(\lambda_t); \overline{\theta}))^{\theta}(1 - l_{fe}(k_t, \tanh^{-1}(\lambda_t); \overline{\theta}))^{1-\theta}\right)^{1-\tau}}{c(k_t, \tanh^{-1}(\lambda_t); l_{fe}(k_t, \tanh^{-1}(\lambda_t); \overline{\theta}))},$$

$k_{t+1} = k'(k_t, \tanh^{-1}(\lambda_t); l_{fe}(k_t, \tanh^{-1}(\lambda_t); \overline{\theta}))$, and $\lambda_{t+1} = \tanh(\rho \tanh^{-1}(\lambda_t) + \sqrt{2}\sigma v_{t+1})$.

To bound the capital we fix an ex-ante upper bound $\overline{k}$, picked sufficiently high that it will bind only with an extremely low probability. As a consequence, the Euler equation (6) implies the residual equation:

$$R(k_t, \lambda_t; \theta) = \frac{\beta}{\sqrt{\pi}} \int_{-1}^{1} \left[\frac{U_{c,t+1}}{U_{c,t}} \left(1 + \alpha \widehat{\lambda}_{t+1} k_{t+1}^{\alpha-1} l_{fe}(k_{t+1}, \tanh^{-1}(\lambda_{t+1}))^{1-\alpha} + \delta\right)\right] \times \exp(-v_{t+1}^2)\, \mathrm{d}v_{t+1} - 1.$$

Now, we define $\Omega = [0, \overline{k}] \times [-1, 1]$ as the domain of $l_{fe}(k, \tanh^{-1}(\lambda); \overline{\theta})$ and divide Ω into non-overlapping rectangles $[k_i, k_{i+1}] \times [\lambda_j, \lambda_{j+1}]$, where k_i is the ith grid point for capital and λ_j is jth grid point for the technology shock. Clearly $\Omega = \cup_{i,j} [k_i, k_{i+1}] \times [\lambda_j, \lambda_{j+1}]$. Each of these rectangles is called an element. The elements may be of unequal size. In our computations we have small elements in the areas of Ω where the economy spends most of the time, while just a few big elements cover wide areas infrequently visited.[6]

Next, we set $\Psi_{ij}(k, \lambda) = \widehat{\Psi}_i(k) \widetilde{\Psi}_j(\lambda)\ \forall i, j$, where

$$\widehat{\Psi}_i(k) = \begin{cases} \dfrac{k - k_{i-1}}{k_i - k_{i-1}} & \text{if } k \in [k_{i-1}, k_i], \\ \dfrac{k_{i+1} - k}{k_{i+1} - k_i} & \text{if } k \in [k_i, k_{i+1}], \\ 0 & \text{elsewhere,} \end{cases} \qquad \widetilde{\Psi}_j(\lambda) = \begin{cases} \dfrac{\lambda - \lambda_{j-1}}{\lambda_j - \lambda_{-1j}} & \text{if } \lambda \in [\lambda_{j-1}, \lambda_j], \\ \dfrac{\lambda_{j+1} - \lambda}{\lambda_{j+1} - \lambda_j} & \text{if } \lambda \in [\lambda_j, \lambda_{j+1}], \\ 0 & \text{elsewhere,} \end{cases}$$

are the basis functions. Note that $\Psi_{ij}(k, \lambda) = 0$ if $(k, \lambda) \notin [k_{i-1}, k_{i+1}] \times [\lambda_{j-1}, \lambda_{j+1}]\ \forall i, j$, i.e., the function is 0 everywhere except inside four elements. Also,

[6]There is a whole area of research concentrated on the optimal generation of an element grid. See Thomson et al. (1985).

$l_{fe}(k_i, \tanh^{-1}(\lambda_j); \overline{\theta}) = \overline{\theta}_{ij}\ \forall i,j$, i.e., the values of $\overline{\theta}$ specify the values of l_{fe} at the corners of each subinterval $[k_i, k_{i+1}] \times [\lambda_j, \lambda_{j+1}]$.

A natural criterion for finding the $\overline{\theta}$ unknowns is to minimize this residual function over the state space given some weight function. A Galerkin scheme implies that we weight the residual function by the basis functions and solve the system of equations:

$$\int_{[0,\overline{k}]\times[-1,1]} \Psi_{i,j}(k,\lambda) R(k,\lambda;\overline{\theta})\,\mathrm{d}k\,\mathrm{d}\lambda = 0 \quad \forall i,j \tag{7}$$

on the $\overline{\theta}$ unknowns.

Since the basis functions are zero outside their element, we can rewrite (7) as:

$$\int_{[k_{i-1},k_i]\times[\lambda_{j-1},\lambda_j]\cup[k_i,k_{i+1}]\times[\lambda_j,\lambda_{j+1}]} \Psi_{i,j}(k,\lambda) R(k,\lambda;\overline{\theta})\,\mathrm{d}k\,\mathrm{d}\lambda = 0 \quad \forall i,j. \tag{8}$$

We evaluate the integrals in (8) using Gauss–Legendre (Press et al., 1992). Since we specify 71 unequal elements in the capital dimension and 31 on the λ axis, we have an associated system of 2201 nonlinear equations. We solve this system with a Quasi-Newton algorithm. The solution delivers our desired policy function $l_{fe}(k, \tanh^{-1}(\lambda); \overline{\theta})$, from which we can find all the other variables in the economy.[7]

3.2.2. *Spectral (Chebyshev polynomials) method*

Like finite elements, spectral methods (Judd, 1992) search for a policy function of the form $l_{\mathrm{sm}}(k,z;\overline{\theta}) = \sum_{i,j} \overline{\theta}_{ij} \Psi_{ij}(k,z)$ where $\Psi_{ij}(k,z)$ is a set of basis functions and $\overline{\theta}$ is a vector of parameters to be determined. The difference with respect to the finite elements is that the basis functions are (almost everywhere) nonzero for most of the state space.

Spectral methods have two advantages over finite elements. First, they are easier to implement. Second, since we can handle a large number of basis functions, the accuracy of the solution is potentially high. The main drawback of the procedure is that, since the policy functions are nonzero for most of the state space, if the policy function displays a rapidly changing local behavior, or kinks, the scheme may deliver a poor approximation.

A common choice for the basis functions are Chebyshev polynomials. Since the domain of Chebyshev polynomials is $[-1,1]$, we need to bound both capital and technology and define the linear map from those bounds into $[-1,1]$. Capital must belong to the set $[0,\overline{k}]$, where $\overline{k}$ is picked sufficiently high that it will bind with an extremely low probability. The bounds for the technological shock, $[\underline{z},\overline{z}]$, come from Taychen's (1986) method to approximate to an AR(1) process. Then, we set $\Psi_{ij}(k,z) = \widehat{\Psi}_i(\Phi_k(k))\widetilde{\Psi}_j(\Phi_z(z))$ where $\widehat{\Psi}_i(\cdot)$ and $\widetilde{\Psi}_j(\cdot)$ are Chebyshev polynomials[8]

[7]Policy function iteration (Miranda and Helmberger, 1988) is a particular case of finite elements when we pick a collocation scheme in the points of a grid, linear basis functions, and an iterative scheme to solve for the unknown coefficients. Experience from numerical analysis shows that nonlinear solvers (as our Newton scheme) or multigrid schemes outperform iterative algorithms (see Briggs et al., 2000). Also Galerkin weightings are superior to collocation for finite elements (Boyd, 2001).

[8]These polynomials can be recursively defined by $T_0(x) = 1$, $T_1(x) = 1$, and for general n, $T_{n+1}(x) = 2T_n(x) - T_{n-1}(x)$. See Boyd (2001) for details.

and $\Phi_k(k)$ and $\Phi_z(z)$ define the linear mappings from $[0,\overline{k}]$and $[\underline{z},\overline{z}]$, respectively to $[-1,1]$.

As in the finite elements method, we use the two Euler equations with the budget constraint substituted in to get a residual function:

$$R(k_t,z_t;\overline{\theta}) = \frac{\beta}{\sqrt{2\pi\sigma}}\int_{-\infty}^{\infty}\left[\frac{U_{c,t+1}}{U_{c,t}}\left(1+\alpha e^{z_{t+1}}k_{t+1}^{\alpha-1}l_{\mathrm{sm}}(k_{t+1},z_{t+1})^{1-\alpha}-\delta\right)\right] \times \exp\left(-\frac{\varepsilon_{t+1}^2}{2\sigma^2}\right)\mathrm{d}\varepsilon_{t+1}, \tag{9}$$

where

$$U_{c,t} = \frac{\left(c(k_t,z_t;l_{\mathrm{sm}}(k_t,z_t;\overline{\theta}))^{\theta}(1-l_{\mathrm{sm}}(k_t,z_t;\overline{\theta}))^{1-\theta}\right)^{1-\tau}}{c(k_t,z_t;l_{\mathrm{sm}}(k_t,z_t;\overline{\theta}))},$$

$k_{t+1}=k'(k_t,z_t;l_{\mathrm{sm}}(k_t,z_t;\overline{\theta}))$, and $z_{t+1}=\rho z_t+\varepsilon_{t+1}$.

Instead of a Galerkin weighting, computational experience (Fornberg, 1998) suggests that, for spectral methods, a collocation (also known as pseudospectral) criterion delivers the best trade-off between accuracy and the ability to handle a large number of basis functions. The points $\{k_i\}_{i=1}^{n_1}$ and $\{z_j\}_{j=1}^{n_2}$ are called the collocation points. We choose the roots of the n_1th order Chebyshev polynomial as the collocation points for capital.[9] This choice is called orthogonal collocation, since the basis functions constitute an orthogonal set. These points are attractive because by the Chebyshev interpolation theorem, if an approximating function is exact at the roots of the n_1th order Chebyshev polynomial, then, as $n_1\to\infty$, the approximation error becomes arbitrarily small. For the technology shock we use Tauchen's finite approximation to an AR(1) process to obtain n_2 points. We also employ the transition probabilities implied by this approximation to compute the integral in Eq. (9).

Therefore, we need to solve the following system of $n_1\times n_2$ equations:

$$R(k_i,z_j;\overline{\theta})=0 \quad \text{for } \forall i,j \text{ collocation points} \tag{10}$$

with $n_1\times n_2$ unknowns $\overline{\theta}_{ij}$. This system is easier to solve than (7), since we will have in general fewer equations and we avoid the integral induced by the Galerkin weighting.[10]

To solve the system we use a quasi-Newton method and an iteration based on the increment of the number of basis functions and a nonlinear transformation of the

[9]The roots are given by $k_i=(x_i+1)/2$, where

$$x_i=\cos\left\{\frac{\pi[2(n_1-i+1)-1]}{2n_1}\right\},\quad i=1,\ldots,n_1.$$

[10]Parametrized expectations (see Marcet and Lorenzoni, 1999 for a description) is a spectral method that uses monomials (or exponents of) in the current states of the economy and Monte Carlo integration. Since monomials are highly collinear and determinist integration schemes are preferred for low dimensional problems over Monte Carlos (Geweke, 1996), we stick with Chebyshev polynomials as our favorite basis.

objective function (Judd, 1992). First, we solve a system with only three collocation points for capital (and n_2 points for the technology shock). Then, we use that solution as a guess for a system with one more collocation point for capital (with the new coefficients being guessed equal to zero). We find a new solution, and continue the procedure until we use up to 11 polynomials in the capital dimension and 9 in the productivity axis.

3.3. Value function iteration

Finally we solve the model using value function iteration. Since the dynamic algorithm is well known we only present a sparse discussion.

Consider the following Bellman operator:

$$TV(k,z) = \max_{c>0,0<l<1,k'>0} \frac{(c^\theta(1-l)^{1-\theta})^{1-\tau}}{1-\tau} + \beta EV(k',z'|z),$$

$$c + k' = \exp(z)k^\alpha l^{1-\alpha} + (1-\delta)k,$$

$$z' = \rho z + \varepsilon.$$

To solve the Bellman operator, we define a grid on capital, $G_k \equiv \{k_1, k_2, \ldots, k_M\}$, and use Taychen's (1986) method to the stochastic process z, $G_z \equiv \{z_1, z_2, \ldots, z_N\}$, and Π^N being the resulting transition matrix with generic element $\pi^N_{i,j} \equiv \Pr(z' = z_j | z' = z_i)$. However, we use those points *only* as a grid for productivity and to compute the expectation of the value function in the next period. When we simulate the model, we interpolate along the productivity dimension.

The algorithm to iterate on the value function for a given grid is given by:

I. Set $n = 0$ and

$$V_0(k,z) = \frac{\left(c_{ss}^\theta (1-l_{ss})^{1-\theta}\right)^{1-\tau}}{1-\tau} \quad \text{for all } k \in G_k \text{ and all } z \in G_z.$$

II. Set $i = 1$.

a. Set $j = 1$ and $r = 1$.

b. 1. Set $s = r$ and $U^s_{i,j} = -\inf$.

2. Use Newton method to find l_s that solves

$$(1-\alpha)\exp(z_j)k_i^\alpha l^{-\alpha}(1-l) + k' = \frac{1-\theta}{\theta}(\exp(z_j)k_i^\alpha l^{1-\alpha} + (1-\delta)k_i - k').$$

3. Compute

$$U^s_{i,j} = \frac{\left(((1-\alpha)\exp(z_j)k_i^\alpha l_s^{-\alpha}(1-l_s))^\theta (1-l_s)^{1-\theta}\right)^{1-\tau}}{1-\tau} + \beta \sum_{r=1}^{N} \pi^N_{j,r} V_n(k_s, z_r).$$

4. If $U_{i,j}^{s-1} \leqslant U_{i,j}^{s}$, then $s \rightsquigarrow s+1$ and go to 2.
5. Define

$$U(k;k_i,z_j) = \frac{\left(((1-\alpha)\exp(z_j)k_i^{\alpha}l^{-\alpha}(1-l))^{\theta}(1-l)^{1-\theta}\right)^{1-\tau}}{1-\tau} + \beta\sum_{r=1}^{N}\pi_{j,r}^{N}\widehat{V}_n(k,z_r)$$

for $k \in [k_{s-2}, k_s]$, where l solves

$$(1-\alpha)\exp(z_j)k_i^{\alpha}l^{-\alpha}(1-l) = \frac{1-\theta}{\theta}(\exp(z_j)k_i^{\alpha}l^{1-\alpha} + (1-\delta)k_i - k_s)$$

and $\widehat{V}_n(k,z_r)$ is computed using interpolation.[11]

6. Let $k_{i,j}^{*} = \operatorname{argmax} U(k;k_i,z_j)$.
7. Set r such that $k_{i,j}^{*} \in [k_r, k_{r+1}]$ and $V_{n+1}(k_i,z_j) = U(k_{i,j}^{*};k_i,z_j)$.

c. If $j<N$, then $j \rightsquigarrow j+1$ and go to b.

III. If $i<N$, $i \rightsquigarrow i+1$ and go to a.

IV. If $\sup_{i,j}|V_{n+1}(k_i,z_j) - V_n(k_i,z_j)|/V_n(k_i,z_j) \geqslant 1.0\mathrm{e}^{-8}$, then $n \rightsquigarrow n+1$ and go to II.[12]

To accelerate convergence, we follow Chow and Tsitsiklis (1991). We start iterating on a small grid. Then, after convergence, we add more points to the grid, and recompute the Bellman operator using the previously found value function as an initial guess (with linear interpolation to fill the unknown values in the new grid points). Iterating with this grid refinement, we move from an initial 8000-point grid into a final one with one million points (25000 points for capital and 40 for the productivity level).

4. Calibration: benchmark case and robustness

To make our comparison results as useful as possible, we pick a benchmark calibration and we explore how those results change as we move to different 'unrealistic' calibrations.

We select the benchmark calibration values for the model as follows. The discount factor $\beta = 0.9896$ matches an annual interest rate of 4 percent (see McGrattan and Prescott, 2000 for a justification of this number based on their measure of the return on capital and on the risk-free rate of inflation-protected U.S. Treasury bonds). The risk aversion $\tau = 2$ is a common choice in the literature. $\theta = 0.357$ matches labor supply to 31 percent of available time in the steady state. We set $\alpha = 0.4$ to match labor share of national income (after the adjustments to national income and product accounts suggested by Cooley and Prescott, 1995). The depreciation rate

[11] We interpolate using linear, quadratic, and Schumaker's splines (Judd and Solnick, 1994). Results were very similar with all three methods because the final grid was so fine that how interpolation was done did not really matter. The results in the paper are those with linear interpolation.

[12] We also monitored convergence in the policy function that was much quicker.

Table 1
Calibrated parameters

Parameter	β	τ	θ	α	δ	ρ	σ
Value	0.9896	2.0	0.357	0.4	0.0196	0.95	0.007

Table 2
Sensitivity analysis

Case	$\sigma = 0.007$	$\sigma = 0.035$
$\tau = 2$	Benchmark	Intermediate Case 3
$\tau = 10$	Intermediate Case 1	Intermediate Case 4
$\tau = 50$	Intermediate Case 2	Extreme

$\delta = 0.0196$ fixes the investment/output ratio. Values of $\rho = 0.95$ and $\sigma = 0.007$ match the stochastic properties of the Solow residual of the U.S. economy. The chosen values are summarized in Table 1.

To check robustness, we repeat our analysis for five other calibrations. Thus, we study the relative performance of the methods both for a nearly linear case (the benchmark calibration) and for highly non-linear cases (high variance/high risk aversion). We increase the risk aversion to 10 and 50 and the standard deviation of the productivity shock to 0.035. Although below we concentrate on the results for the benchmark and the extreme case, the intermediate cases are important to make sure that our comparison across calibrations does not hide nonmonotonicities. Table 2 summarizes our different cases.

Also, we briefly discuss some results for the deterministic case $\sigma = 0$, since they well help us understand some characteristics of the proposed methods, for the case $\tau = 1$ (log utility function), and for lower β's.

5. Numerical results

In this section we report our numerical findings. We concentrate on the benchmark and extreme calibrations, reporting the intermediate cases when they clarify the argument. First, we present and discuss the computed policy functions. Second, we show some simulations. Third, we perform the χ^2 accuracy test proposed by Den Haan and Marcet (1994), we report the Euler equation errors as in Judd (1992) and Judd and Guu (1997). Fourth, we study the robustness of the results. Finally, we discuss implementation and computing time.

5.1. Policy functions

One of our first results is the policy functions. We plot the decision rules for labor supply when $z = 0$ over a capital interval centered on the steady state level

of capital for the benchmark calibration in Fig. 1 and for investment in Fig. 2. Similar figures could be plotted for other values of z. We omit them because of space considerations.

Since many of the nonlinear methods provide indistinguishable answers, we observe only four lines in both figures. Labor supply is very similar in all methods, especially in the neighborhood of 23.14, the steady state level of capital. Only far away from that neighborhood can we appreciate differences. A similar description applies to the policy rule for investment except for the loglinear approximation where the rule is pushed away from the other ones for low and high capital. The difference is big enough that even the monotonicity of the policy function is lost. We must be cautious, however, mapping differences in choices into differences in utility. The Euler error function below provides a better view of the welfare consequences of different approximations.

Bigger differences appear as we increase risk aversion and the variance of the shock. The policy functions for the extreme calibration are presented in Figs. 3 and 4. In these figures we change the interval reported because, owing to the risk aversion/high variance of the calibration, the equilibrium paths fluctuate around higher levels of capital (between 30 and 45) when the solution method accounts for risk aversion (i.e., all the nonlinear ones).

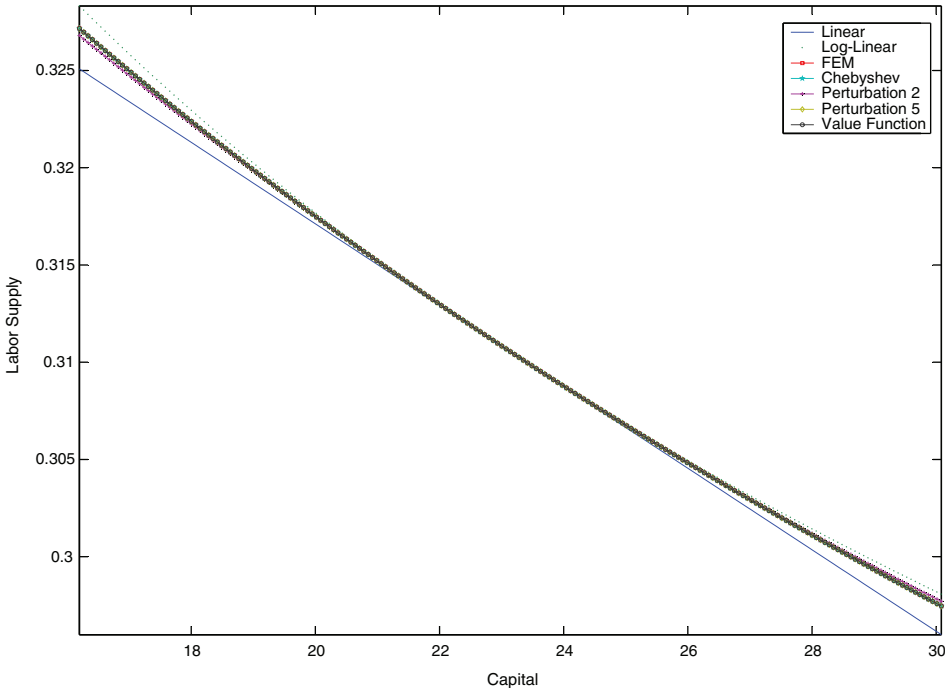

Fig. 1. Labor supply at $z = 0$, $\tau = 2/\sigma = 0.007$.

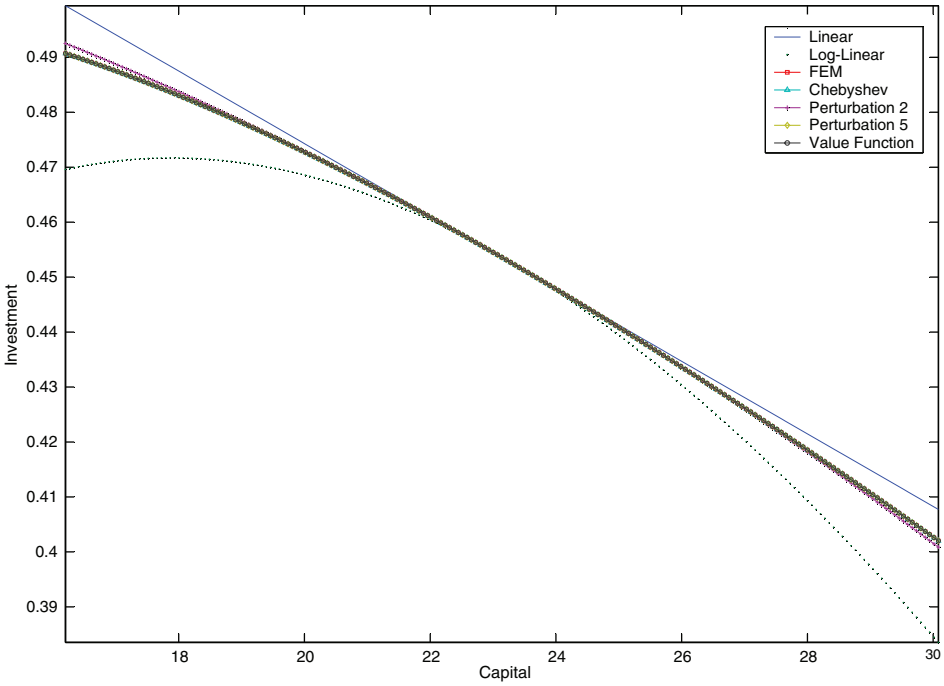

Fig. 2. Investment at $z = 0$, $\tau = 2/\sigma = 0.007$.

We highlight several results. First, the linear and loglinear policy functions deviate from all the other ones: they imply much less labor (around 10 percent) and investment (up to 30 percent) than nonlinear methods. This difference in level is due to the lack of correction for increased variance of the technology shock by these two approximations, since they are certainty-equivalent. Second, just correcting for quadratic terms in the second order perturbation allows us to get the right level of the policy functions. This is a key argument in favor of phasing out linearizations and substituting at least second order perturbations for them. Third, the policy function for labor and investment approximated by the fifth order perturbation changes from concavity into convexity for values of capital bigger than 45 (contrary to the theoretical results). This change of slope will cause problems below in our simulations. Fourth, the policy functions have a positive slope because of precautionary behavior. We found that the change in slope occurs for τ, around 40.

5.2. Simulations

Practitioners often rely on statistics from simulated paths of the economy. We computed 1000 simulations of 500 observations each for all methods. To make comparisons meaningful we kept the productivity shock constant across methods for each particular simulation.

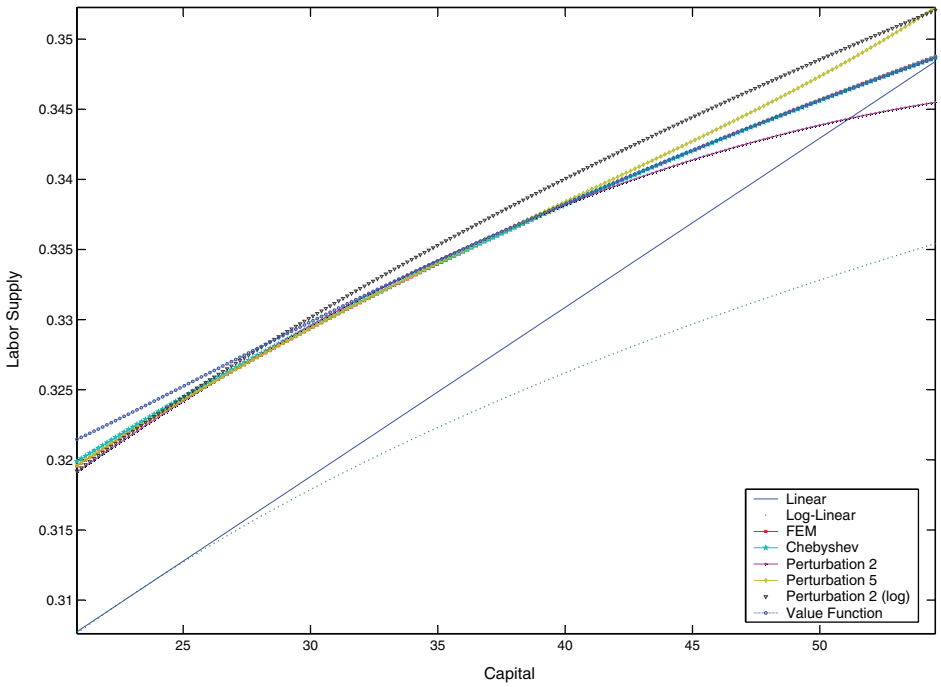

Fig. 3. Labor supply at $z = 0$, $\tau = 50/\sigma = 0.035$.

For the benchmark calibration, the simulation from all the models generates nearly identical equilibrium paths, densities of the variables, and business cycle statistics. These results are a simple consequence of the similarity of the policy functions. Because of space considerations, we do not include these results, but they are available at the companion web page at http://www.econ.upenn.edu/~jesusfv/companion.htm.

More interesting is the case of the extreme calibration. We plot in Figs. 5–7 the histograms of output, capital, and labor for each solution method. In these histograms we see three groups: first, the two linear methods, second, the perturbations, and finally the three global methods (value function, finite elements, and Chebyshev). The last two groups have the histograms shifted to the right: much more capital is accumulated and more labor supplied by all the methods that allow for corrections by variance. The empirical distributions of nonlinear methods accumulate a large percentage of their mass between 40 and 50, while the linear methods rarely visit that region. Even different nonlinear methods provide quite a diverse description of the behavior of economy. In particular the three global methods are in a group among themselves (nearly on top of each other) separated from perturbations that lack enough variance. Higher risk aversion/high variance also have an impact on business cycle statistics. For example, investment is three

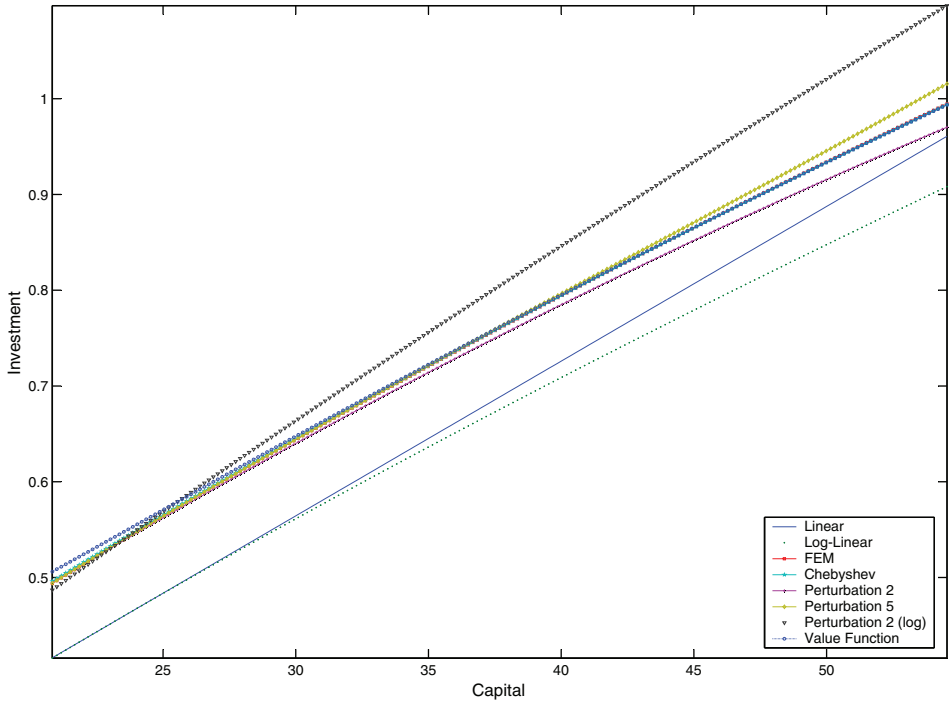

Fig. 4. Investment at $z = 0$, $\tau = 50/\sigma = 0.035$.

times more volatile in the linear simulation than with finite elements despite the filtering of the data.

The simulations show a drawback of using perturbations to characterize equilibrium economies when disturbances are normal. For instance, in 39 simulations out of the 1000 (not shown on the histograms), fifth order perturbation generated a capital that exploded. The reason for that abnormal behavior is the change in the slope of the policy functions reported above. When the economy travels into that part of the policy functions the simulation falls in an unstable path and the results need to be disregarded. Jin and Judd (2002) suggest the use of disturbances with bounded support to solve this problem.

5.3. *A χ^2 accuracy test*

From our previous discussion it is clear that the consequences for simulated equilibrium paths of using different methods are important. A crucial step in our comparison then is the analysis of the accuracy of the computed approximations to figure out which one we should prefer.

We begin that investigation by implementing the χ^2-test proposed by Den Haan and Marcet (1994). The authors noted that if the equilibrium of the economy is

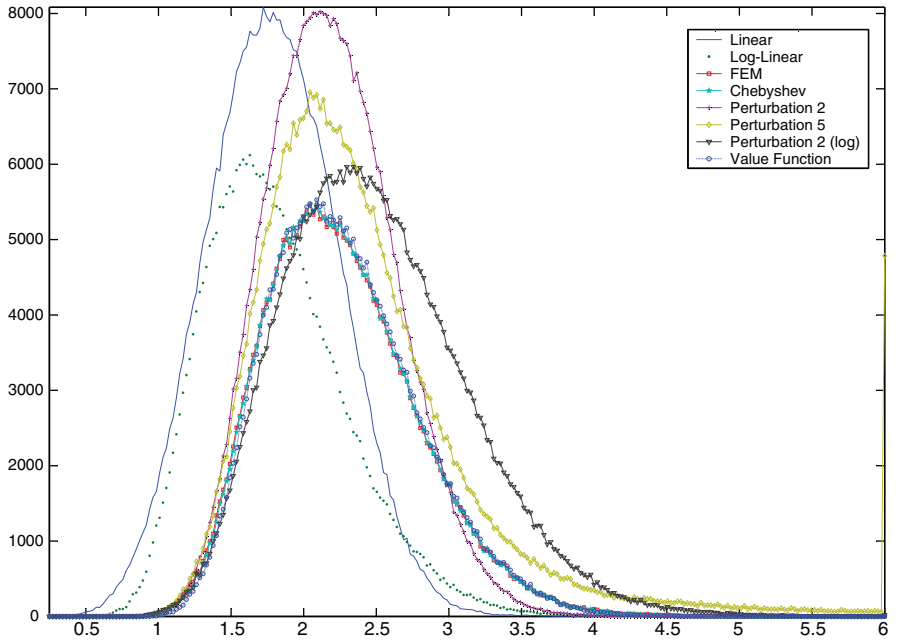

Fig. 5. Density of output, $\tau = 50/\sigma = 0.035$.

characterized by a system of equations $f(y_t) = \mathrm{E}_t(\phi(y_{t+1}, y_{t+2}, ..))$ where the vector y_t contains all the n variables that describe the economy at time t, $f : \Re^n \to \Re^m$ and $\phi : \Re^n \times \Re^\infty \to \Re^m$ are known functions and $\mathrm{E}_t(\cdot)$ represents the conditional expectation operator, then:

$$\mathrm{E}_t(u_{t+1} \otimes h(x_t)) = 0 \tag{11}$$

for any vector x_t measurable with respect to t with $u_{t+1} = \phi(y_{t+1}, y_{t+2}, ..) - f(y_t)$ and $h : \Re^k \to \Re^q$ being an arbitrary function.

Given one of our simulated series of length T from the method i in the previous section, $\{y_t^i\}_{t=1}^T$, we can find $\{u_{t+1}^i, x_t^i\}_{t=1}^T$ and compute the sample analog of (11):

$$B_T^i = \frac{1}{T}\sum_{t=1}^{T} u_{t+1}^i \otimes h(x_t^i). \tag{12}$$

Clearly (12) would converge to zero as T increases almost surely if the solution method were exact. However, given the fact that we only have numerical methods to solve the problem, this may not be the case. However, the statistic $T(B_T^i)'(A_T^i)^{-1}B_T^i$ where A_T^i is a consistent estimate of the matrix $\sum_{t=-\infty}^{\infty} \mathrm{E}_t[(u_{t+1} \otimes h(x_t))(u_{t+1} \otimes h(x_t))']$ given solution method i, converges in distribution to a χ^2 with qm degrees of freedom under the null that (11) holds. Values of the test above the critical value can be interpreted as evidence against the accuracy of the solution.

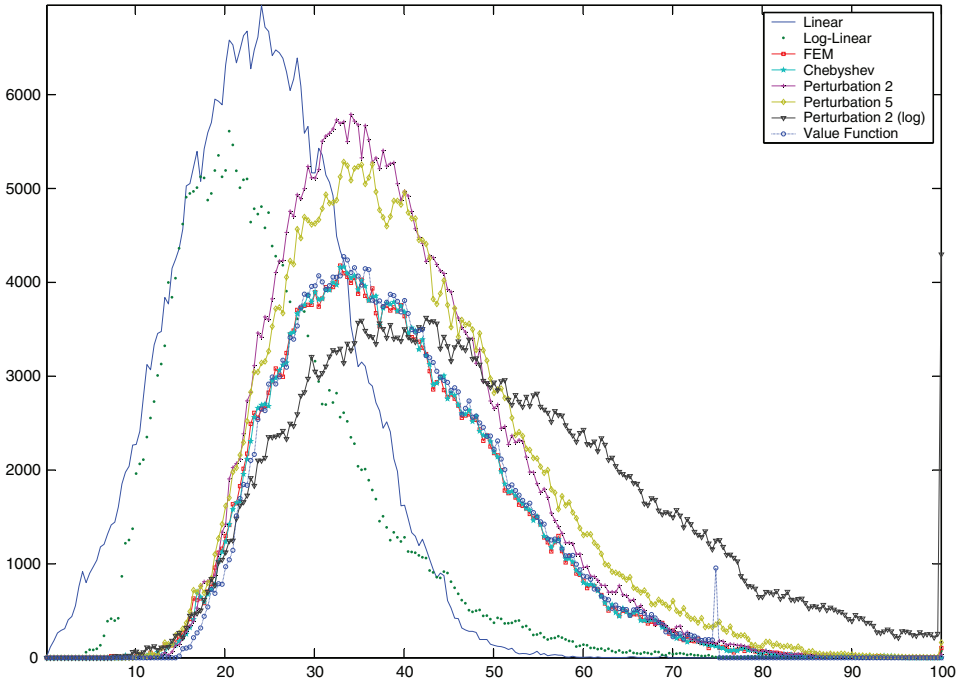

Fig. 6. Density of capital, $\tau = 50/\sigma = 0.035$.

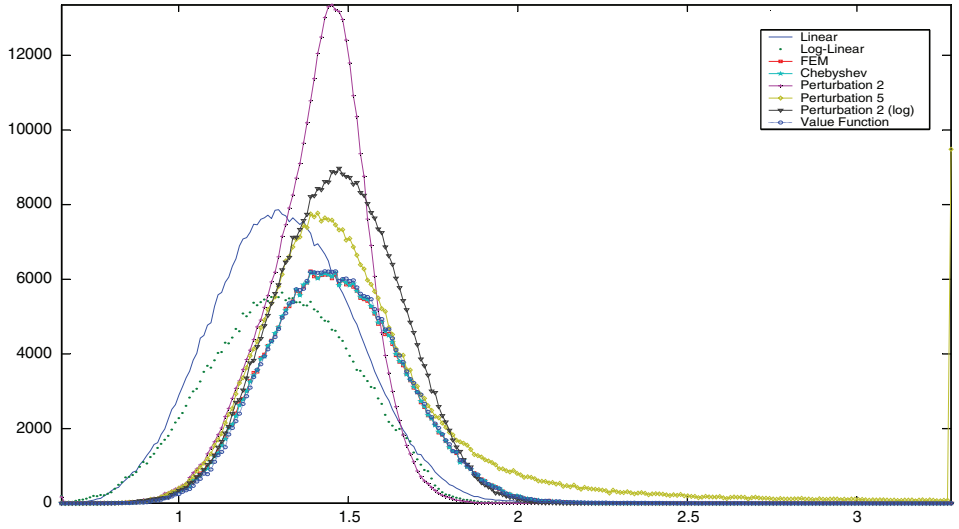

Fig. 7. Density of consumption, $\tau = 50/\sigma = 0.035$.

Table 3
χ^2 Accuracy test, $\tau = 2/\sigma = 0.007$

	Less than 5%	More than 95%
Linear	3.10	5.40
Log-linear	3.90	6.40
Finite elements	3.00	5.30
Chebyshev	3.00	5.40
Perturbation 2	3.00	5.30
Perturbation 5	3.00	5.40
Value function	2.80	5.70

Table 4
χ^2 Accuracy test, $\tau = 50/\sigma = 0.035$

	Less than 5%	More than 95%
Linear	0.43	23.42
Log-linear	0.40	28.10
Finite elements	1.10	5.70
Chebyshev	1.00	5.20
Perturbation 2	0.90	12.71
Perturbation 2-log	0.80	22.22
Perturbation 5	1.56	4.79
Value function	0.80	4.50

Since any solution method is an approximation, as T grows we will eventually reject the null. To control for this problem, we can repeat the test for many simulations and report the percentage of statistics in the upper and lower critical 5 percent of the distribution. If the solution provides a good approximation, both percentages should be close to 5 percent.

We report results for the benchmark calibration in Table 3 (the Empirical CDF can be found at the companion web page).[13] All the methods perform similarly and reasonably close to the nominal coverages, with a small bias toward the right of the distribution. Also, and contrary to some previous findings for simpler models (Den Haan and Marcet, 1994; Christiano, 1990) it is not clear that we should prefer loglinearization to linearization.

We present the results for the extreme case in Table 4.[14] Now the performance of the linear methods deteriorates enormously, with unacceptable coverages (although again linearization in levels is no worse than loglinearization). On the other hand, nonlinear methods deliver a good performance, with very reasonable coverages on

[13]We use a constant, k_t, k_{t-1}, k_{t-2} and z_t as our instruments, 3 lags and a Newey–West estimator of the matrix of variances–covariances (Newey and West, 1987).

[14]The problematic simulations as described above are not included in these computations.

the upper tail (except second order perturbations). The lower tail behavior is poor for all methods.

5.4. Euler equation errors

The previous test is a simple procedure to evaluate the accuracy of a solution. That approach may suffer, however, from three problems. First, since all methods are approximations, the test will display low power. Second, orthogonal residuals can be compatible with large deviations from the optimal policy. Third, the model will spend most of the time in those regions where the density of the stationary distribution is higher. However, sometimes it is important to ensure accuracy far away from the steady state.

Judd (1992) proposes to determine the quality of the solution method defining normalized Euler equation errors. First, note that in our model the intertemporal condition:

$$u_c'(c(k_t,z_t),l(k_t,z_t)) = \beta \mathrm{E}_t\{u_c'(c(k(k_t,z_t),z_{t+1}),l(k(k_t,z_t),z_{t+1}))R(k_t,z_t,z_{t+1})\}, \tag{13}$$

where $R(k_t,z_t,z_{t+1}) = (1+\alpha e^{z_{t+1}}k(k_t,z_t)^{\alpha-1}l(k(k_t,z_t),z_{t+1})^{1-\alpha}-\delta)$ is the gross return rate of capital, should hold exactly for given k_t, and z_t. Since the solution methods used are only approximations, (13) will not hold exactly when evaluated using the computed decision rules. Instead, for solution method i with associated policy rules $c^i(\cdot,\cdot)$, $l^i(\cdot,\cdot)$, and $k^i(\cdot,\cdot)$, and the implied gross return of capital $R^i(k_t,z_t,z_{t+1})$, we can define the Euler equation error function $EE^i(\cdot,\cdot)$ as

$$EE^i(k_t,z_t) \equiv 1 - \frac{\left(\dfrac{\beta \mathrm{E}_t\left\{u_c'\left(c^i(k^i(k_t,z_t),z_{t+1}),l^i(k^i(k_t,z_t),z_{t+1})\right)R^i(k_t,z_t,z_{t+1})\right\}}{\theta(1-l^i(k^i(k_t,z_t),z_{t+1}))^{(1-\theta)(1-\tau)}}\right)^{1/(\theta(1-\tau)-1)}}{c^i(k_t,z_t)}.$$

This function determines the (unit free) error in the Euler equation as a fraction of the consumption given the current states k_t, and z_t and solution method i. Judd and Guu (1997) interpret this error as the relative optimization error incurred by the use of the approximated policy rule. For instance, if $EE^i(k_t,z_t) = 0.01$, then the agent is making a \$1 mistake for each \$100 spent. In comparison, $EE^i(k_t,z_t) = 1e^{-8}$ implies that the agent is making a 1 cent mistake for each one million dollars spent.

The Euler equation error is also important because we know that, under certain conditions, the approximation error of the policy function is of the same order of magnitude as the size of the Euler equation error. Correspondingly, the change in welfare is of the square order of the Euler equation error (Santos, 2000).

Plots of the Euler equation error functions can be found at the companion web page. To get a better view of the relative performance of each approximation and since plotting all the error functions in the same plot is cumbersome, Figs. 8 and 9

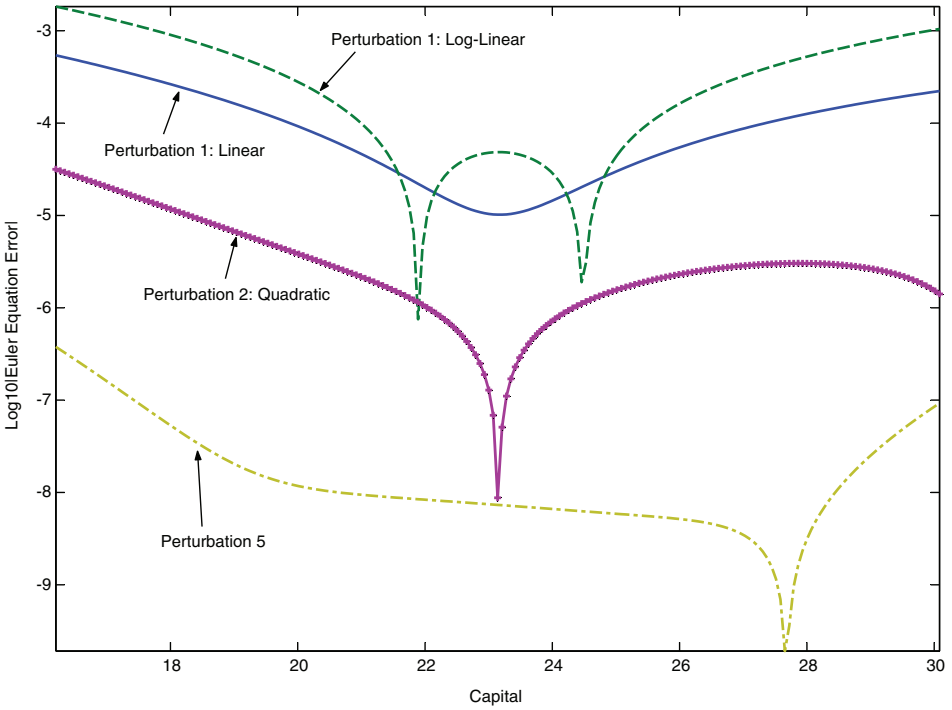

Fig. 8. Euler equation errors at $z = 0$, $\tau = 2/\sigma = 0.007$.

display a transversal cut of the errors when $z = 0$. We report the absolute errors in base 10 logarithms to ease interpretation. A value of -3 means \$1 mistake for each \$1000, a value of -4 a \$1 mistake for each \$10000, and so on. Also, we separate the results in two figures for clarity. In Fig. 8, we include all perturbation methods (first, second, and fifth order), while, in Fig. 9, we plot finite elements, Chebyshev polynomials, and value function iteration plus a linear approximation for comparison purposes.

In the figures, we can see how the loglinear approximation is worse than the linearization except at two valleys where the error in levels goes from positive into negative values. Finite elements and Chebyshev polynomials perform three orders of magnitude better than linear methods. Perturbations' accuracy is even more impressive. Other transversal cuts at different technology levels reveal similar patterns.

We can summarize the information from Euler equation error functions in two complementary ways. First, following Judd and Guu (1997), we report the maximum error in a set around the steady state. We pick a square given by capital between 70 percent and 130 percent of the steady state (23.14) and for a range of technology shocks from -0.065 to 0.065 (with zero being the level of technology in the

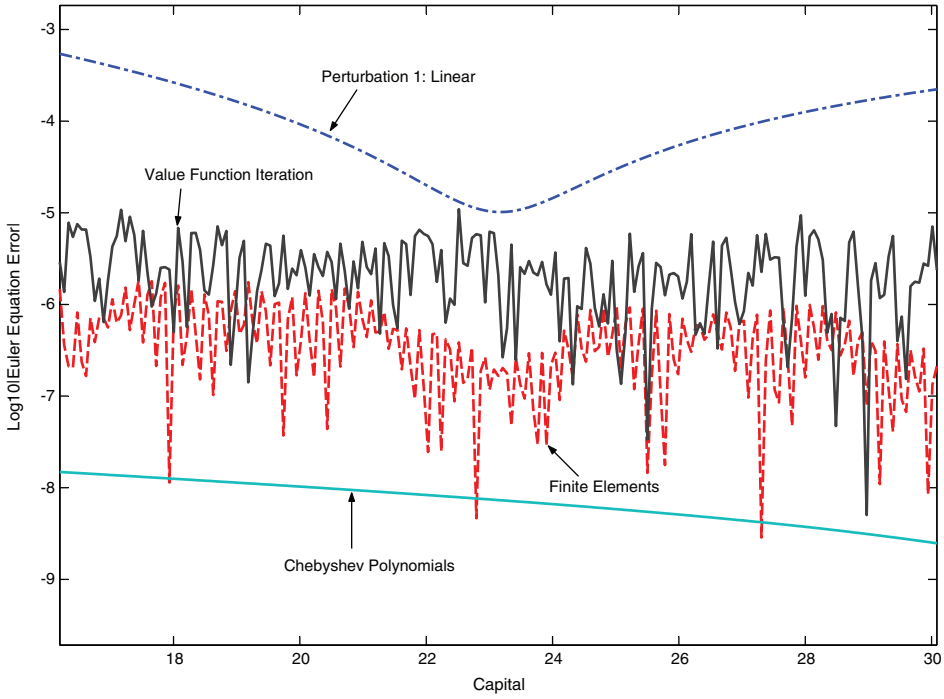

Fig. 9. Euler equation errors at $z = 0$, $\tau = 2/\sigma = 0.007$.

deterministic case).[15] The maximum Euler error is useful as a measure of accuracy because it bounds the mistake that we are incurring owing to the approximation. Also, the literature on numerical analysis has found that maximum errors are good predictors of the overall performance of a solution.

Table 5 presents the maximum Euler equation error for each solution method. We can see how there are three levels of accuracy. Linear and loglinear, between -2 and -3, the different perturbation and projection methods, all around -3.3, and value function around -4.43. This table can be read as suggesting that, for this benchmark calibration, all methods display acceptable behavior, with loglinear performing the worst of all and value function the best.

The second procedure to summarize Euler equation errors is to combine them with the information from the simulations to find the average error. This exercise is a generalization of the Den Haan–Marcet test where, instead of using the conditional expectation operator, we estimate an unconditional expectation using the population

[15]0.065 corresponds to roughly the 99.5th percentile of the normal distribution given our parameterization. The interval for capital includes virtually 100 percent of the stationary distributions as computed in the previous subsection. Varying the interval for capital changes the size of the maximum Euler error but not the relative ordering of the errors induced by each solution method.

Table 5
Euler errors (Abs($\log_{10}$))

	Max Euler error	Integral of the Euler errors
Linear	−2.8272	−4.6400
Log-linear	−2.2002	−4.2002
Finite elements	−3.3801	−5.2700
Chebyshev	−3.3281	−5.4330
Perturbation 2	−3.3138	−5.3179
Perturbation 5	−3.3294	−5.4330
Value function	−4.4343	−5.6498

distribution. This integral is a welfare measure of the loss induced by the use of the approximating method. Results are also presented in Table 5. We use the distribution from value function iteration. Since the distributions are nearly identical for all methods, the table is also nearly the same if we integrate with respect to any other distribution.

The two sets of numbers in Table 5 show that linearization in levels must be preferred over loglinearization for the benchmark calibration. The problems of linearization are not as much due to the presence of uncertainty but to the curvature of the exact policy functions. Even with no uncertainty, the Euler equation errors of the linear methods (not reported here) are very poor in comparison with the nonlinear procedures.

We repeat our exercise for the extreme calibration. Figs. 10 and 11 display results for the extreme calibration $\tau = 50$, $\sigma = 0.035$, and $z = 0$ (again we have changed the capital interval to make it representative). This shows the huge errors of the linear approximation in the relevant parts of the state space. The plot is even worse for the loglinear approximation. Finite elements still displays robust and stable behavior over the state space. The local definition of the basis functions picks the strong nonlinearities induced by high risk aversion and high variance. Chebyshev's performance is also very good and delivers similar accuracies. The second and fifth order perturbations keep their ground and perform relatively well for a while but then, around values of capital of 40, they strongly deteriorate. Value function iteration delivers an uniformly high accuracy.

These findings are reinforced by Table 6. Again we report the absolute max Euler error and the integral of the Euler equation errors computed as in the benchmark calibration (except the bigger window for capital).[16] From the table we can see three clear winners (finite elements, Chebyshev, and value function) and a clear loser (loglinear) with the other results in the middle. The performance of loglinearization is disappointing. The max Euler error implies an error of \$1 for each \$27 spent. In

[16]As before, we use the stationary distribution of capital from value function iteration. The results with any of the other two global non-linear methods are nearly the same.

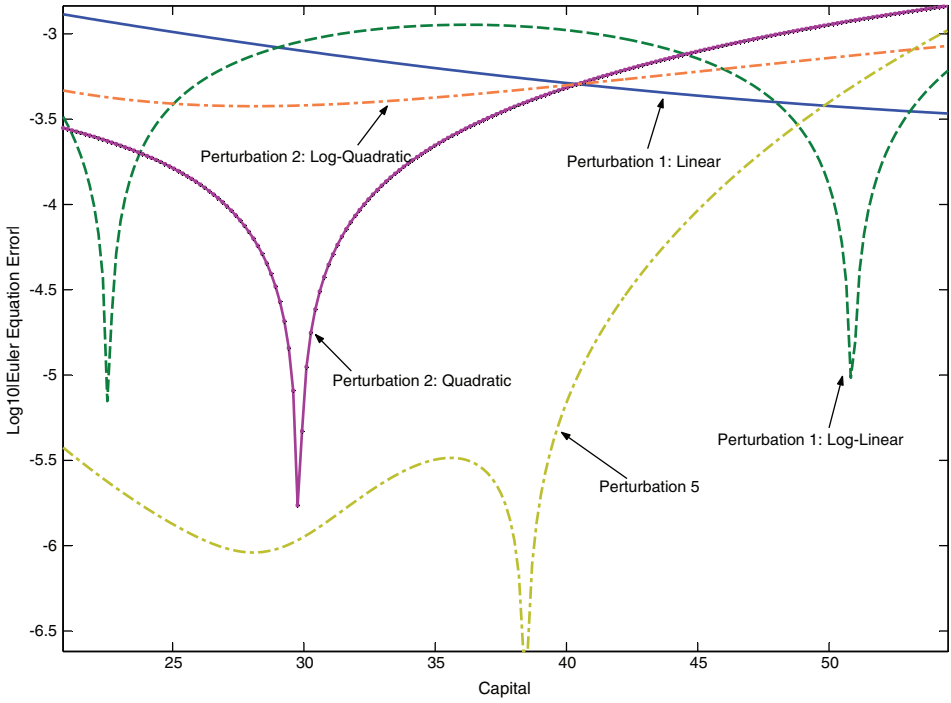

Fig. 10. Euler equation errors at $z = 0$, $\tau = 50/\sigma = 0.035$.

comparison, the maximum error of the linearization is \$1 for each \$305. The poor performance of the perturbation is due to the quick deterioration of the approximation outside the range of capital between 20 and 45.

5.5. *Robustness of results*

We explored the robustness of our results with respect to changes in the parameter values. Because of space constraints, we comment only on four of these robustness exercises, although we perform a few more experiments.

A first robustness exercise was to evaluate the four intermediate parameterizations described above. The main lesson from those four cases was that they did not uncover any nonmonoticity of the Euler equation errors. As we moved, for example, toward higher risk aversion, the first order perturbations began to deteriorate while non-linear methods maintained their high accuracy.

A second robustness exercise was to reduce to zero the variance of the productivity shock, i.e., to make the model deterministic. The main conclusion was that first order perturbation still induced high Euler equation errors, while the non-linear methods delivered Euler equation errors that were close to machine zero along the central parts of the state space.

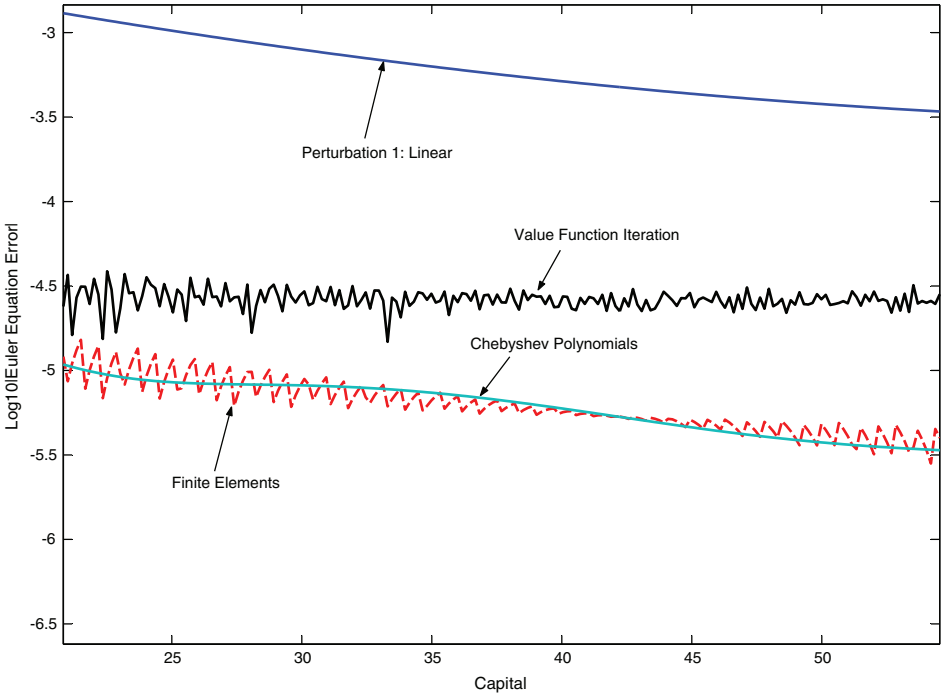

Fig. 11. Euler equation errors at $z = 0$, $\tau = 50/\sigma = 0.035$.

A third robustness exercise was to change the utility function to a log form. The results in this case were very similar to our benchmark calibration. This is not surprising. Risk aversion in the benchmark case was 1.357,[17] while in the log case it is 1. This small difference in risk aversion implies small differences in policy rules and approximation errors between the benchmark calibration and the log case. With log utility linearization had a maximum Euler error of -2.8798 and loglinearization of -2.0036. This was one of the only cases where loglinearization did better than linearization. The non-linear methods were all hovering around -3.3 as in the benchmark case (for example, finite elements was -3.3896, Chebyshev -3.3435, second order perturbation -3.3384, and so on).

A fourth robustness exercise was to reduce the discount factor, β, to 0.98 to generate an steady state annual interest rate of 8.5 percent. This exercise checks the behavior of the solution methods in economies with high return to capital. Some economists (Feldstein, 2000) have argued that high interest rates are a better description of the data than the lower 4 percent commonly used in quantitative exercises in macro. Our choice of 8.5 percent is slightly above the upper bound of Feldstein's computations for 1946–1995. The results in this case are also very similar

[17]Given our utility function with leisure, the Arrow–Pratt coefficient of relative risk aversion is $1 - \theta(1 - \tau)$. The calibrated values of $\tau = 1$ and $\theta = 0.357$ imply the risk aversion in the text.

Table 6
Euler errors (Abs($\log_{10}$))

	Absolute max Euler error	Integral of the Euler errors
Linear	−1.4825	−4.1475
Log-linear	−1.4315	−2.6131
Finite elements	−2.8852	−4.4685
Chebyshev	−2.5269	−4.6578
Perturbation 2	−1.9206	−3.1101
Perturbation 5	−1.9104	−3.0501
Perturbation 2 (log)	−1.7724	−3.1891
Value function	−4.015	−4.4949

to the benchmark case. First order perturbations cause maximum Euler errors between −2 and −3 and the nonlinear methods around −3.26. The relative size and ordering of errors are also the same.

We conclude from our robustness analysis that the lessons learned in this section are likely to hold for a large region of parameter values.

5.6. Implementation and computing time

We briefly discuss implementation and computing time. Traditionally (for example, Taylor and Uhlig, 1990), computational papers have concentrated on the discussion of the running times. Being an important variable, sometimes running times are of minor relevance in comparison with programming and debugging time. A method that may run in a fraction of a second but requires thousands of lines of code may be less interesting than a method that takes a minute but has a few dozen lines of code. Of course, programming time is a much more subjective measure than running time, but we feel that some comments are useful. In particular, we use lines of code as a proxy for the implementation complexity.[18]

The first order perturbation (in level and in logs) takes only a fraction of a second in a 1.7 MHz Xeon PC running Windows XP (the reference computer for all times below), and it is very simple to implement (less than 160 lines of code in Fortran 95 with generous comments). Similar in complexity is the code for the higher order perturbations, around 64 lines of code in *Mathematica* 4.1, although *Mathematica* is much less verbose. The code runs in between 2 and 10 s depending on the order of the expansion. This observation is the basis of our comment the marginal cost of perturbations over linearizations is close to zero. The finite elements method is perhaps the most complicated method to implement: our code in Fortran 95 has above 2000 lines and requires some ingenuity. Running time is moderate, around 20 min, starting from conservative initial guesses and a slow update. Chebyshev

[18]Unfortunately, *Matlab*'s and Fortran 95's inability to handle higher order perturbations stops us from using only one programming language. We use Fortran 95 for all other methods because of speed considerations.

polynomials are an intermediate case. The code is much shorter, around 750 lines of Fortran 95. Computation time varies between 20 s and 3 min, but it requires a good initial guess for the solution of the system of equations. Finally, value function iteration code is around 600 lines of Fortran 95, but it takes between 20 and 250 h to run.[19]

6. Conclusions

In this paper we have compared different solution methods for dynamic equilibrium economies. We have found that higher order perturbation methods are an attractive compromise between accuracy, speed, and programming burden, but they suffer from the need to compute analytical derivatives and from some instabilities. In any case they must clearly be preferred to linear methods. If such a linear method is required (for instance, if we want to apply the Kalman filter), the results suggest that it is better to linearize in levels than in logs. The finite elements method is a robust, solid method that conserves its accuracy over a long range of the state space and different calibrations. Also, it is perfectly suited for parallelization and estimation purposes (Fernández-Villaverde and Rubio-Ramírez, 2004). However, it is costly to implement and moderately intensive in running time. Chebyshev polynomials share most of the good properties of finite elements if the problem is as smooth as ours and they may be easier to implement. However it is nor clear that this result will generalize to less well-behaved applications.

We finish by pointing to several lines of future research. First, the results in Williams (2004) suggest that further work integrating the perturbation method with small noise asymptotics are promising. Second, it can be fruitful to explore newer nonlinear methods such as the adaptive finite element method (Verfürth, 1996), the weighted extended *B*-splines finite element approach (Höllig, 2003), and element-free Galerkin methods (Belytschko et al., 1996) that improve on the basic finite elements method by exploiting local information and error estimator values.

Acknowledgements

We thank Kenneth Judd for encouragement and criticisms, Jonathan Heathcote, José Victor Ríos-Rull, Stephanie Schmitt-Grohé, and participants at several seminars for useful comments. Mark Fisher helped us with *Mathematica*. Jesús Fernández-Villaverde thanks the NSF for financial support under the project SES-0338997. Beyond the usual disclaimer, we must note that any views expressed herein

[19]The exercise of fixing computing time and evaluating the accuracy of the solution delivered by each method in that time is not very useful. Perturbation is in a different class of time requirements than finite elements and value function iteration (with Chebyshev somewhere in the middle). Either we set such a short amount of time that the results from finite elements and value function iteration are meaningless, or the time limit is not binding for perturbations and again the comparison is not informative.

are those of the authors and not necessarily those of the Federal Reserve Bank of Atlanta or the Federal Reserve System.

References

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P., 1996. Meshless methods: an overview and recent developments. Computer Methods in Applied Mechanics and Engineering 139, 3–47.

Bender, C.M., Orszag, S.A., 1999. Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory. Springer, New York, Inc., New York.

Benítez-Silva, H., Hall, G., Hitsch, G.J., Pauletto, G., Rust, J., 2000. A comparison of discrete and parametric approximation methods for continuous-state dynamic programming problems. Mimeo, SUNY at Stony Brook.

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

Boyd, J.P., 2001. Chebyshev and Fourier Spectral Methods, second ed. Dover Publications, Mineola.

Briggs, W.L., Henson, V.E., McCormick, S.F., 2000. A Multigrid Tutorial, second ed. Society for Industrial and Applied Mathematics, Philadelphia.

Christiano, L.J., 1990. Linear-quadratic approximation and value-function iteration: a comparison. Journal of Business Economics and Statistics 8, 99–113.

Christiano, L.J., Fisher, J.D.M., 2000. Algorithms for solving dynamic models with occasionally binding constraints. Journal of Economic Dynamics and Control 24, 1179–1232.

Chow, C.S., Tsitsiklis, J.N., 1991. An optimal one-way multigrid algorithm for discrete-time stochastic control. IEEE Transaction on Automatic Control 36, 898–914.

Cooley, T.F., Prescott, E.C., 1995. Economic growth and business cycles. In: Cooley, T.F. (Ed.), Frontiers of Business Cycle Research. Princeton University Press, Princeton, pp. 1–38.

Den Haan, W.J., Marcet, A., 1994. Accuracy in simulations. Review of Economic Studies 61, 3–17.

Feldstein, M., 2000. The distributional effects of an investment-based social security system. NBER Working Paper 7492.

Fernández-Villaverde, J., Rubio-Ramírez, J.F., 2004. Estimating macroeconomic models: a likelihood approach. Federal Reserve Bank of Atlanta Working Paper 2004-1.

Fornberg, B., 1998. A Practical Guide to Pseudospectral Methods. Cambridge University Press, Cambridge.

Gaspar, J., Judd, K., 1997. Solving large-scale rational-expectations models. Macroeconomic Dynamics 1, 45–75.

Geweke, J., 1996. Monte Carlo simulation and numerical integration. In: Amman, H., Kendrick, D., Rust, J. (Eds.), Handbook of Computational Economics. Elsevier-North Holland, Amsterdam.

Hall, R., 1971. The dynamic effects of fiscal policy in an economy with foresight. Review of Economic Studies 38, 229–244.

Höllig, K., 2003. Finite Element Methods with *B*-Splines. Society for Industrial and Applied Mathematics, Philadelphia.

Hughes, T.R.J., 2000. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publications, Mineola.

Jin, H., Judd, K.L., 2002. Perturbation methods for general dynamic stochastic models. Mimeo, Hoover Institution.

Judd, K.L., 1992. Projection methods for solving aggregate growth models. Journal of Economic Theory 58, 410–452.

Judd, K.L., 1998. Numerical Methods in Economics. MIT Press, Cambridge.

Judd, K.L., Guu, S.M., 1993. Perturbation solution methods for economic growth model. In: Varian, H. (Ed.), Economic and Financial Modelling in Mathematica. Springer, New York Inc., New York.

Judd, K.L., Guu, S.M., 1997. Asymptotic methods for aggregate growth models. Journal of Economic Dynamics and Control 21, 1025–1042.

Judd, K.L., Guu, S.M., 2001. Asymptotic methods for asset market equilibrium analysis. Economic Theory 18, 127–157.

Judd, K.L., Jin, H., 2004. Applying PertSolv to complete market RBC models. Mimeo, Hoover Institution.

Judd, K.L., Solnick, A., 1994. Numerical dynamic programming with shape-preserving splines. Mimeo, Hoover Institution.

King, R.G., Plosser, C.I., Rebelo, S.T., 2002. Production, growth and business cycles: technical appendix. Computational Economics 20, 87–116.

Klein, P., 2000. Using the generalized Schur form to solve a multivariate linear rational expectations model. Journal of Economic Dynamics and Control 24, 1405–1423.

Kydland, F.E., 1989. Monetary policy in models with capital. In: van der Ploeg, F., de Zeuw, A.J. (Eds.), Dynamic Policy Games in Economies. North-Holland, Amsterdam.

Kydland, F.E., Prescott, E.C., 1982. Time to build and aggregate fluctuations. Econometrica 50, 1345–1370.

Magill, J.P.M., 1977. A local analysis of N-sector capital under uncertainty. Journal of Economic Theory 15, 219–221.

Marcet, A., Lorenzoni, G., 1999. The parametrized expectations approach: some practical issues. In: Marimon, R., Scott, A. (Eds.), Computational Methods for the Study of Dynamic Economies. Oxford University Press, Oxford.

Marimón, R., Scott, A., 1999. Computational Methods for the Study of Dynamic Economies. Oxford University Press, Oxford.

McGrattan, E., Prescott, E.C., 2000. Is the stock market overvalued? Quarterly Review 24, 20–40.

Miranda, M.J., Fackler, P.L., 2002. Applied Computational Economics and Finance. MIT Press, Cambridge.

Miranda, M.J., Helmberger, P.G., 1988. The effects of commodity price stabilization programs. American Economic Review 78, 46–58.

Newey, W., West, K.D., 1987. A simple, positive, heteroskedasticity and autocorrelation consistent covariance matrix. Econometrica 55, 703–705.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in Fortran 77: The Art of Scientific Computing. Cambridge University Press, Cambridge.

Rust, J., 1996. Numerical dynamic programming in economics. In: Amman, H., Kendrick, D., Rust, J. (Eds.), Handbook of Computational Economics. Elsevier-North Holland, Amsterdam.

Rust, J., 1997. Using randomization to break the curse of dimensionality. Econometrica 65, 487–516.

Santos, M.S., 1999. Numerical solution of dynamic economic models. In: Taylor, J.B., Woodford, M. (Eds.), Handbook of Macroeconomics, vol. 1a. North-Holland, Amsterdam.

Santos, M.S., 2000. Accuracy of numerical solutions using the Euler equation residuals. Econometrica 68, 1377–1402.

Santos, M.S., Vigo, J., 1998. Analysis of error for a dynamic programming algorithm. Econometrica 66, 409–426.

Schmitt-Grohé, S., Uribe, M., 2004. Solving dynamic general equilibrium models using a second-order approximation to the policy function. Journal of Economic Dynamics and Control 28, 755–775.

Sims, C.A., 2000. Second order accurate solution of discrete time dynamic equilibrium models. Mimeo, Princeton University.

Sims, C.A., 2002. Solving linear rational expectations models. Computational Economics 20, 1–20.

Tauchen, G., 1986. Finite state Markov-chain approximations to univariate and vector autoregressions. Economics Letters 20, 177–181.

Taylor, J.B., Uhlig, H., 1990. Solving nonlinear stochastic growth models: a comparison of alternative solution methods. Journal of Business Economics and Statistics 8, 1–17.

Thomson, J.F., Warsi, Z.U.A., Mastin, C.W., 1985. Numerical Grid Generation: Foundations and Applications. Elsevier, Amsterdam.

Verfürth, R., 1996. Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. Wiley-Teubner, New York.

Williams, N., 2004. Small noise asymptotics for a stochastic growth model. Journal of Economic Theory 119, 271–298.