# A New Algorithm for Solving Dynamic Stochastic Macroeconomic Models[*]

Viktor Dorofeenko
Department of Economics and Finance
Institute for Advanced Studies
Stumpergasse 56
A-1060 Vienna, Austria

Gabriel S. Lee
Department of Economics and Finance
University of Regensburg
Universitaetsstrasse 31
93053 Regensburg, Germany

Kevin D. Salyer (Corresponding Author)
Department of Economics
University of California
Davis, CA 95616
Contact Information:
Salyer: (530) 752 8359; E-mail: kdsalyer@ucdavis.edu

November 2005

**Abstract**

We introduce a new algorithm that can be used to solve stochastic dynamic general equilibrium models. This approach exploits the fact that the equations defining equilibrium can be viewed as set of differential algebraic equations in the neighborhood of the steady-state. Then a modified recursive upwind Gauss Seidel method can be used to determine the global solution. This method, within the context of a standard real business cycle model, is compared to projection, perturbation, and linearization approaches and demonstrated to be fast

[*]We gratefully acknowledge financial support from Jubiläumsfonds der Oesterreichischen Nationalbank (Jubiläumsfondsprojekt Nr. 11042).

and globally accurate. This comparison is done within a discrete state setting with heteroskedasticity in the technology shocks. It is shown that linearization methods perform poorly in this environment even though the unconditional variance of shocks is relatively small.

- *JEL Classification: C63,C68,E37*
- Keywords: *numerical methods, projection methods, real business cycles.*

## 1 Introduction

Modern quantitative macroeconomics necessarily involves the use of numerical methods in order to compute the equilibrium behavior of a model economy. As initially introduced by Magill (1977) and later used by Kydland and Prescott (1982) in their seminal work on business cycle models, linearization methods have been the preferred solution approach. Such methods are easy to implement and, as shown by Christiano (1990), do not introduce significant approximation errors for many settings studied by macroeconomists.

But, as discussed in Judd (2002), linearization methods are not trouble free. Quoting from that paper: “For example, Tesar (1995) uses the Kydland-Prescott method and found an example where completing asset markets will make agents worse off. This result violates general equilibrium theory and can only be attributed to the numerical method used.” (p.2) More recently, Kim and Kim (2003) have shown that this error in welfare analysis is symptomatic of linearized models and argue in favor of second-order approximation methods; variations on this theme have been developed by Sims (2000) and Schmitt-Grohe and Uribe (2004).

In order to study more complicated settings, non-linear methods have also been proposed that employ either projection techniques (Judd (1992) and McGrattan (1999)) or perturbation techniques (Judd and Guu (1997)). In a recent contribution to understanding these approaches, Aruoba, Fernandez-Villaverde, and Rubio-Ramirez (2003) examine the accuracy of these methods (along with traditional linearization and log-linearization methods) within the context of a prototypical real business cycle model. Their results replicate Christiano’s earlier analysis in that, for economies characterized by low risk aversion (i.e. little curvature in the utility function) and shocks that do not push the economy far from the steady-state (i.e. small variance of technology shocks), linearization methods do quite well. (footnote – log linearization is poor.) However, linearization methods, not surprisingly, deteriorate quickly in the presence of large shocks and high risk aversion.

This paper complements and extends the analysis by Aruoba, et al. (2003) in two dimensions. First, we examine discrete state settings so that heteroskedasticity in the technology shock can be introduced. In particular, we examine a crash state scenario and demonstrate that linearization methods perform poorly in this environment. We show that, even though the magnitude of the unconditional variance of the technology shock would lead one to conjecture reasonably small approximation errors (as suggested by Christiano and Aruoba et al.) for linear methods, the volatility of the conditional variances undermines this conjecture. Recent papers by Barro (2005) and Bloom (2005) have argued forcefully for the presence of large shocks to uncertainty in the economy and, hence, our analysis motivates the use of more sophisticated solution methods in such settings.

The second and major contribution of this paper is the introduction of a new algorithm to solve stochastic dynamic economies; for our analysis we use Hansen's (1985) real business cycle model (again, with the shocks following a discrete-state Markov process). Our approach involves two parts: first, a one-pass continuous modification of the Upwind Gauss-Seidel Algorithm (Judd (1998)) is used to solve for the non-stochastic problem, and, second, an implicit iterative scheme is employed to account for the stochastic effects. In the latter iterative approach, the small numerical magnitudes of the stochastic terms (e.g. cross-state transition probabilities in the case of discrete-state Markovian processes or variances for continuous AR processes) produces relatively fast convergence. We will refer below to our method as to the RUGS (Recursive Upwind Gauss-Seidel) method. The algorithm has two strengths: (1) It is computationally fast; and (2) It has high global (i.e. non-local) accuracy. We test the performance and accuracy of our algorithm in comparison with other popular nonlinear methods using the analysis by Aruoba et al. as a template.

In particular, we consider the following methods as comparison tests for our algorithm:

1. A modification of the Value Function Iteration Algorithm (VFI) as it is implemented in Danthine et al. (1989), which is used here mostly to produce a standard time unit for other more advanced methods.

2. A perturbation method based on the Taylor expansion near the deterministic equilibrium point as described in Judd and Jin (2002).

3. A projection method using Chebyshev Polynomials spectral expansion of the sought policy functions. The projection of the residual is performed by the collocation procedure.

4. Standard linearization and log-linearization approximation methods.

We exclude from the benchmark set the Finite Element Method also considered in Aruoba et al., (2003) because, as shown there, in the case of smooth policy functions, it does not do better than the spectral expansion with Chebyshev Polynomials. Another interesting non-local method we don't test here is the Pade approximation considered in Judd and Guu (1997) for a simple deterministic capital growth problem.

## 2 The Benchmark Problem

The benchmark problem for the algorithm is a discrete-state version of the familiar real business cycle model presented in Hansen (1985).

Consider the stochastic optimization problem

$$\max_{c_t, n_t} E_0 \left[ \sum_{t=0}^{\infty} \beta^t u\left(c_t, 1 - n_t\right) \right] \tag{1}$$
$$\textit{subject to } c_t = \lambda_t f\left(k_t, n_t\right) + (1 - \Omega) k_t - k_{t+1}$$

where $c_t, n_t, k_t$, and $\lambda_t$ denote individual consumption, labor hours, capital stock accumulated and technology of production correspondingly; the functions $u(\cdot)$ and $f(\cdot)$ are one-period utility and production function; the constants $\beta$ and $\Omega$ represent agents' discount factor and the depreciation rate of capital. As in Danthine, Donaldson & Mehra (1989), the technology shock, $\lambda_t$, assumes the discrete set of values $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ and follows a Markov process with the transition probability matrix $\mathbf{p}$.

To apply our numerical algorithm, first rewrite eq. (1) as a Bellman Equation. The value function $V(k, \lambda)$ is defined by (with consumption eliminated via the (always binding) resource constraint):

$$V(k, \lambda) = \max_{k_1, n} \left[ U(k, k_1, n, \lambda) + \beta \sum_{\lambda' \in \Lambda} p_{\lambda\lambda'} V\left(k_1, \lambda'\right) \right] \tag{2}$$

with $U(k, k_1, n, \lambda) = u(\lambda f(k, n) + (1 - \Omega) k - k_1, 1 - n)$.

The associated necessary conditions are (with $U_i$ denoting the partial derivative with respect to the $i$th argument):

$$U_2(k, k_1, n, \lambda) + \beta \sum_{\lambda' \in \Lambda} p_{\lambda\lambda'} V_1\left(k_1, \lambda'\right) = 0 \tag{3}$$

$$U_3\left(k,k_1,n,\lambda\right)=0 \tag{4}$$

$$V_1\left(k_1,\lambda\right)=U_1\left(k,k_1,n,\lambda\right) \tag{5}$$

Note that the set of necessary conditions form a complete differential-algebraic system of equations for the sought value function, $V\left(k,\lambda\right)$, and policy functions $k_1\left(k,\lambda\right)$ and $n\left(k,\lambda\right)$. Combining these we have:

$$\begin{gathered}U_2\left(k,k_1\left(k,\lambda\right),n\left(k,\lambda\right),\lambda\right)+\\ \beta\sum_{\lambda'\in\Lambda}p_{\lambda\lambda'}U_1\left[k_1\left(k,\lambda\right),k_1\left(k_1\left(k,\lambda\right),\lambda'\right),n\left(k_1\left(k,\lambda\right),\lambda'\right),\lambda'\right]=0\end{gathered} \tag{6}$$

$$U_3\left(k,k_1\left(k,\lambda\right),n\left(k,\lambda\right),\lambda\right)=0 \tag{7}$$

## 3 The Algorithm

### 3.1 Solving the Deterministic Problem

Consider first the steady-state system of the economy in which $\lambda=1$; then eqs. $(6),(7)$ become:

$$U_2\left(k,k_1\left(k\right),n\left(k\right)\right)+\beta U_1\left(k_1\left(k\right),k_1\left(k_1\left(k\right)\right),n\left(k_1\left(k\right)\right)\right)=0 \tag{8}$$

$$U_3\left(k,k_1\left(k\right),n\left(k\right)\right)=0 \tag{9}$$

As is well known, the presence of the nested terms $k_1\left(k_1\left(k\right)\right),n\left(k\left(k_1\right)\right)$ in the above equations implies that, in general, the solution involves functional methods. However, certain properties of the solution permit the treatment of eqs. $(8),(9)$ as algebraic and find its solution by a standard algorithm (see sections 12.3 and 12.5 in the monograph of Judd, (1998) for the description of the Upwind Gauss-Seidel (UGS) method used here and an example of its application to a continuous-state deterministic Bellman equation). This is demonstrated below:

The solution of the system defined by eqs. $(8),(9)$ which maximizes the right-hand side of the Bellman equation has the unique stationary point $k_1\left(k_s\right)=k_s$ that satisfies the equations

$$U_2\left(k_s,k_s,n_s\right)+\beta U_1\left(k_s,k_s,n_s\right)=0 \tag{10}$$

$$U_3\left(k_s,k_s,n_s\right)=0 \tag{11}$$

In equations $(10,11)$ we introduce the corresponding stationary value of labor $n_s=n(k_s)$. The stationary point can be found by applying a standard non-linear equation solution method to the above equations.

Given this solution, the stability and uniqueness of the steady-state implies the inequalities:

$$\begin{aligned} &\forall k > k_s : k_1(k) < k \\ &\forall k < k_s : k_1(k) > k \end{aligned} \tag{12}$$

Assume that we already have the solution to the system defined by the eqs. $(8), (9)$ over the interval $[k_s, k_c]$. That is, over this interval, we know the functions $k_1(k) = \tilde{k}_1(k); n(k) = \tilde{n}(k)$; and now we extend the interval to $[k_s, k_r]$ where $k_r > k_c$. The first inequality in eq.(12) implies that for some interval to the right of $k_c$, i.e. $k_c < k < k_c + \delta$, the value of the sought function lies to the left of $k_c$; therefore the nested functions $k_1(k_1(k)), n(k_1(k))$ of such $k \in [k_c, k_c + \delta]$ may be calculated using the known functions $\tilde{k}(k), \tilde{n}(k)$. Then, for this interval, the system of eqs. $(8), (9)$ takes the form:

$$\begin{gathered} U_2(k, k_1(k), n(k)) + \beta U_1\left(k_1(k), \tilde{k}_1(k_1(k)), \tilde{n}(k_1(k))\right) = 0 \\ U_3(k, k_1(k), n(k)) = 0 \end{gathered} \tag{13}$$

Note, critically, that this system of equations does not involve nested functions so it can be treated as an algebraic equation and solved by an appropriate standard numerical method.

Obtaining in this way the solution over the interval $[k_c, k_c + \delta]$, it is possible to extend the solution interval to the right; recursive repetition of the procedure can be done until the desired boundary, $k_r$, is reached. Thus choosing the right point of initial interval infinitely close to the stationary point , we can then step by step extend the solution to the endpoint . Clearly, the second inequality in eq. (12) allows us to apply the same procedure to the left of $k_s$. This procedure generates the solution over the entire interval $[k_l, k_r]$.

This algorithm allows one to solve the deterministic growth problem using a fast one-pass algorithm. Note that the described procedure employs the proper ordering of values of the state variable, $k$, and starts from the absorbing state $k_s$; hence by the classification of (Judd, 1998), it can be treated as a continuous modification of the Upwind Gauss-Seidel Algorithm. Note that, unlike the example presented in the above Judd's monograph, where the UGS method is used to solve the Bellman equation for the Value function, we apply it here to the necessary conditions to obtain the policy functions and thus avoiding the time consuming maximization procedure.

### 3.2 Extending to Stochastic Settings

The procedure described above cannot be trivially generalized to the stochastic problem defined by eq. (2), because

1. The stochastic problem has multiple stationary points, depending on the current value of technological factor.
2. The inequalities in eq.(12) may simultaneously have opposite signs for different values of $\lambda$.

These properties significantly complicate and slow down the algorithm when extended in a straightforward manner to the general problem. Consequently, we will use a modification of a simple iterative scheme that employs the one-pass algorithm described above. As it is demonstrated below, this scheme converges quickly to the desired accuracy, so the time of calculation does not grow considerably.

Since the conditional probabilities in any state $i$ sum to unity, rewrite eqs. $(6),(7)$ as:

$$U_2\left[k,k_1(k,\lambda),n(k,\lambda),\lambda\right]+\beta U_1\left[k_1(k,\lambda),k^{(\lambda,\lambda)}(k),n^{(\lambda,\lambda)}(k),\lambda\right]=$$
$$\beta\sum_{\lambda'\in\Lambda}p_{\lambda\lambda'}\left(U_1\left[k_1(k,\lambda),k^{(\lambda,\lambda)}(k),n^{(\lambda,\lambda)}(k),\lambda\right]-U_1\left[k_1(k,\lambda),k^{\left(\lambda,\lambda'\right)}(k),n^{\left(\lambda,\lambda'\right)}(k),\lambda'\right]\right) \tag{14}$$

$$U_3\left[k,k_1(k,\lambda),n(k,\lambda),\lambda\right]=0 \tag{15}$$

where the following notation is used for expositional purposes:

$$k^{\left(\lambda,\lambda'\right)}(k)=k_1\left(k_1(k,\lambda),\lambda'\right);n^{\left(\lambda,\lambda'\right)}(k)=n\left(k_1(k,\lambda),\lambda'\right) \tag{16}$$

Note that the left hand side of the first equation in eq. (14) is simply the deterministic case described in the section above. Hence, we will find the solution to the stochastic setting by an implicit iterative method which solves the two equations in eqs. $(14),(15)$. This solution is, of course, the solution to the original problem given in eq. (2). The iterative method can be summarized as:

$$LHS\left(k_{m+1},n_{m+1},k_{m+1}^{(\lambda,\lambda)},n_{m+1}^{(\lambda,\lambda)},\lambda\right)=RHS\left(k_m,n_m,...\right) \tag{17}$$

where $k_m = k_1^{(m)}(k, \lambda), n_m = n^{(m)}(k, \lambda)$ represent the $m$th iteration of the policy functions and $LHS$ and $RHS$ refer to eqs. $(14), (15)$. The system of equations described above is heterogeneous modification of the deterministic problem described in the previous section and, consequently, can be solved using the one-pass method described there. Also, since the right-hand side of eq. (14) contains the sum of addends proportion to $p_{\lambda\lambda'}$ with zero diagonal terms, we expect fast convergence for this scheme since the non-diagonal elements are typically small given the high persistence typically assumed for the technology shock.

Thus the complete scheme of the numeric solution of the stochastic growth problem consists of the following steps:

1. For each value of $\lambda_i$, solve the correspondent deterministic problem (defined by eqs.$(8)$, and $(9)$) with the one-pass UGS method in order to obtain the policy functions $k_0^{(\lambda_i,\lambda_i)} = k_1^{(0)}(k, \lambda_i), n_0^{(\lambda_i,\lambda_i)} = n^{(0)}(k, \lambda_i)$ . Use them as the initial guess in the iterative algorithm.

2. Repeat the iterations defined by eq. (17) until the desired accuracy is reached.

Each of these steps requires an application of the above one-pass algorithm.

In addition to the expected speed of convergence, the other advantage of this scheme is that equations (17) are always solved separately, for each value of $\lambda$, so the time of computation grows only linearly with the number of discrete states considered. The cost of this speedup is the necessity to use iterative process with some number of steps. This iterative process may also diverge under some circumstances and for some initial guesses and that may sometimes narrow the applicability region of the method.

## 4 Comparison of Algorithms

In this section we compare the performance of the proposed algorithm (RUGS) with the performance of the four numerical methods (linearization, projection, perturbation, and value function iteration (VFI)) mentioned in the introduction. The brief description of these methods is presented in the next section.

Each computational method is used to solve the basic problem given in eq. (1); for all simulations we hold constant the first order autocorrelation and unconditional standard deviation of the technology shock. Specifically, we assume that $Corr\left(\lambda_t, \lambda_{t-1}\right) = 0.95$ and $\sigma_\varepsilon = 0.007$. These are standard values used in the literature. For the linear, log-linear, and perturbation methods, we assume that the technology shock follows a continuous $AR(1)$ process. For the RUGS, VFI, and Projection method, we assume that $\lambda$ follows a discrete state Markov process. We examine two settings: a two-state process in which the conditional second moments of $\lambda_t$ are constant and a five state process with a low probability crash state. This latter setting introduces significant variation in the conditional standard deviation of $\lambda_t$. Consequently, the role of non-linearities in the policy rules should be highlighted in this setting.

For the two-state process, the transition probability matrix and possible realizations are given by:

$$p = \begin{pmatrix} 0.97 & 0.03 \\ 0.03 & 0.97 \end{pmatrix}, \Lambda = (0.98, 1.02) \tag{18}$$

For the five-state process, the transition probability matrix and possible realizations are given by:

$$p = \begin{pmatrix} 1-2\Delta_1 & \Delta_1/2 & \Delta_1/2 & \Delta_1 & 0 \\ \Delta_1 & 1-2\Delta_1-\Delta_2-\delta_1 & \Delta_2 & \Delta_1 & \delta_1 \\ \Delta_1 & \Delta_2 & 1-2\Delta_1-\Delta_2-\delta_2 & \Delta_1 & \delta_2 \\ \Delta_1 & \Delta_1/2 & \Delta_1/2 & 1-2\Delta_1 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix} \tag{19}$$

$$\Lambda = (1-\delta\lambda, 1, 1, 1+\delta\lambda, 1-\Delta\lambda)$$

$$\Delta_1 = 0.017, \Delta_2 = 0.2, \delta_1 = 0.005, \delta_2 = 0.01, \delta\lambda = 0.027, \Delta\lambda = 0.35$$

As mentioned above, these representations have the same unconditional moments for $\lambda_t$.

The production function is assumed to be Cobb-Douglas

$$f\left(k_t, n_t\right) = k_t^{\alpha} h_t^{1-\alpha} \tag{20}$$

Utility takes the functional form:

$$U\left(c_t, 1-h_t\right) = \frac{\left(c^{\theta}\left(1-h\right)^{1-\theta}\right)^{1-\tau} - 1}{1-\tau} \tag{21}$$

For all simulations, the following parameter values were used:

| Parameter | $\beta$ | $\Omega$ | $\alpha$ | $\tau$ | $\theta$ | $\rho$ | $\sigma$ |
|---|---|---|---|---|---|---|---|
| Value | 0.96 | 0.1 | 0.36 | 2; 8 | 0.344 | 0.95 | 0.007 |

These again are common values and produce steady-state values for the capital output ratio and time spent in work activity consistent with U.S. data. The models were solved under the assumption of low ($\tau = 2$) and high ($\tau = 8$) values of relative risk aversion.

For each of the methods, we compare accuracy and speed of convergence. For accuracy, we follow Aruoba et al. (2004) and define accuracy in terms of the Euler equation residual. Using the relationship $c = [u_c(c, 1-n)]^{\frac{1}{\theta(1-\tau)-1}}$ , the Euler equation (EE) error is given by:

$$EE(k,\lambda) = 1 - \left[ -\beta \frac{\sum_{\lambda' \in \Lambda} p_{\lambda\lambda'} U_1 \left[ k_1(k,\lambda), k_1\left(k_1(k,\lambda), \lambda'\right), n\left(k_1(k,\lambda), \lambda'\right), \lambda' \right]}{U_2\left(k, k_1(k,\lambda), n(k,\lambda), \lambda\right)} \right]^{\frac{1}{\theta(1-\tau)-1}} \tag{22}$$

We examine the Maximal and Average Euler Equation Error. Maximal EE error is defined by:

$$\max EE = \max_{k \in [k_l, k_r], \lambda \in \Lambda} EE(k,\lambda) \tag{23}$$

where the interval $[k_l, k_r]$ equals to the solution interval or, for linear and perturbation methods, it equals to the narrower interval of attraction (for the definition see Appendix). The average EE error is defined as average of absolute value of $EE(k,\lambda)$ over a time series sample generated using the tested policy functions. We take the length of sample of 30000 to avoid the dependence on a specific realization.

## 4.1 Speed of convergence

We turn first to an analysis of the speed of the algorithms. (We do not report these for the linear and log-linear procedures since these are virtually instantaneous.) Table 1 presents the results (time is measured in seconds) for the remaining procedures. (We only report these for the case where relative risk aversion is equal to 2; the results for the high risk aversion economies were virtually identical.) Note that in the five state economy, we compare only the projection and RUGS methods since the perturbation approach is not appropriate (it assumes the technology shock is homoskedastic) and the value function method is too time consuming.

Table 1: Speed of Convergence

| Method | Time ($n = 2$) | Time ($n = 5$)) |
|---|---|---|
| Perturbation (5th order polynomial) | 82 | *na* |
| Value Function Iteration | 77 | *na* |
| Projection | 17 | 199 |
| RUGS | 4.1 | 25 |

Hence, we see that the RUGS approach is faster than all methods due to the one-pass aspect of the procedure; moreover, this relative superior performance becomes greater in higher dimensional settings.

## 4.2 Error

The maximal and average Euler equation errors for each procedure are given in Tables 2 and 3. Note that the figures given are the negative of the actual (logarithmic) values. These numbers represent the percentage cost in term of steady-state consumption due to the approximation. A value of 2, for example, implies a mistake of \$1 for every \$100 spent while a value of 4 implies a \$1 mistake for every \$10,000 spent.

Table 2: Euler Equation Errors in the Low Risk Aversion Economy ($\tau = 2$)

| | $n = 2$ | | $n = 5$ | |
|---|---|---|---|---|
| | *Max EE* | *Average EE* | *Max EE* | *Average EE* |
| Linear | 1.7 | 3.9 | *na* | *na* |
| Log-linear | 1.9 | 3.2 | *na* | *na* |
| Perturbation | 2.2 | 9.3 | *na* | *na* |
| Value Function Iteration | 2.2 | 3.2 | *na* | *na* |
| Projection | 7.1 | 7.8 | 6.7 | 7.4 |
| RUGS | 6.6 | 6.7 | 6.9 | 7.5 |

Table 3: Euler Equation Errors in the Low Risk Aversion Economy ($\tau = 8$)

| | $n = 2$ | | $n = 5$ | |
|---|---|---|---|---|
| | *Max EE* | *Average EE* | *Max EE* | *Average EE* |
| Linear | 1.7 | 4.0 | *na* | *na* |
| Log-linear | 1.9 | 3.2 | *na* | *na* |
| Perturbation | 2.8 | 9.0 | *na* | *na* |
| Value Function Iteration | 2.2 | 3.1 | *na* | *na* |
| Projection | 7.6 | 8.3 | 6.7 | 7.9 |
| RUGS | 6.5 | 6.6 | 6.9 | 7.5 |

It is clear from the numbers reported in both Tables that the projection method and RUGS are roughly the same but significantly more accurate than the other procedures.

As a final comparison, the policy rules for investment generated by the procedures are plotted as a function of capital (with the technology shock held constant at the steady-state (or unconditional mean) value of unity). Note that the linear and log-linear procedures are relatively accurate in the homoskedastic environment but this accuracy deteriorates when the technology shock is assumed to have a crash state.

## 5 Brief Description of Numerical Procedures

We briefly discuss the principles and methods used for the other numerical procedures used for the comparison to the proposed RUGS method. For more details, see the noted references; also, the reader is referred to Aruoba, et al. where a nice summary of these procedures is also presented.

### 5.1 Projection method (spectral collocation with Chebyshev Polynomials)

References: Judd (1996), Judd (1998), Boyd (1989).

This method uses the representation of the sought policy functions as a spectral series of Chebyshev Polynomials with finite number of terms (in other words, we project the sought functions on the finite set of orthogonal

Chebyshev Polynomials of the 1st kind):

$$k_1(k,\lambda)=\sum_{i=0}^{M}\kappa_i(\lambda)\,T_i(z(k)) \tag{24}$$
$$n(k,\lambda)=\sum_{i=0}^{M}\nu_i(\lambda)\,T_i(z(k))$$

With $z(k)$ defined as:

$$z(k)=2\left[\frac{k-(k_{\max}+k_{\min})/2}{k_{\max}-k_{\min}}\right] \tag{25}$$

The linear transformation in eq.(25) makes the transition to the Chebyshev Polynomial domain $[-1,1]$ . The use of eqs.$(24),(25)$ into the Euler equations eqs. $(6),(7)$ results in approximate relations of the type:

$$F\left(\sum_{i=0}^{M}\kappa_i(\lambda)\,T_i(z(k)),\sum_{i=0}^{M}\nu_i(\lambda)\,T_i(z(k)),z,\lambda\right)=0 \tag{26}$$

with the sought coefficients $\kappa_i(\lambda),\nu_i(\lambda)$. The simplest procedure projecting the left-hand side of eq.(25) to a finite basis is the collocation procedure, when eq.(25) is assumed to be exact at some finite set of points $z=z_1,...,z_{M+1}$ (that choice corresponds to the projection basis of Dirac-delta functions). The usual choice of $(M+1)$ zeros of Chebyshev Polynomial $T_{M+1}(z)$ as collocation points provides the smallest (in certain sense) deviation of the function $F$ from zero between the points. So, finally, the set of equations for the sought coefficients has the form:

$$F\left(\sum_{i=0}^{M}\kappa_i(\lambda)\,T_i(z_j),\sum_{i=0}^{M}\nu_i(\lambda)\,T_i(z_j),z_j,\lambda\right)=0;\quad j=(1,...,M+1) \tag{27}$$

where $z_j=\cos\left[\pi\frac{2j-1}{2(M+1)}\right]$; $j=(1,...,M+1)$.
The solution of eq.(27) using numerical methods for nonlinear algebraic equations yields the required result.

Note finally, that the above collocation procedure with Chebyshev Polynomials produces the coefficients $\kappa_i(\lambda),\nu_i(\lambda)$, which numerically coincides with those obtained by the projection of eq. (26) on the set of Chebyshev Polynomials themselves (Galerkin scheme) within the error of approximation.

### 5.1.1 Perturbation Method

References: Judd & Guu (1997); Judd & Jin (2002)

This method exploits an ordinary Taylor series expansion of the sought functions near the deterministic steady-state point:

$$k_1(k,\lambda) = k_s + \sum_{i=1}^{M}\sum_{j=0}^{i} a_j^i (k-k_s)^j (\lambda-1)^{i-j} \tag{28}$$
$$n(k,\lambda) = n_s + \sum_{i=1}^{M}\sum_{j=0}^{i} b_j^i (k-k_s)^j (\lambda-1)^{i-j}$$

where $k_s, n_s$ denote the steady-state values.

The Taylor expansion of eqs. $(6),(7)$ using substitution eq.(28) leads to recurrent equations for the series coefficients $\left(a_j^i, b_j^i\right)$, $j=0,...,i$, given $\left(a_j^l, b_j^l\right)$, $l=1,...,i-1$ . As shown in (Judd & Jin, 2002), the equations for $i \geq 2$ are linear and under certain additional conditions have a unique solution. The equations for $i=1$ correspond to a linear approximation and, in our case, are quadratic in $a_1^1$. We choose the solution which corresponds to the contracting mapping $\left|a_1^1\right| < 1$ and, thereby, implies that we are on the stable arm of the saddle path.

### 5.1.2 Value Function Iteration

References: Judd (1998), Danthine, Donaldson, & Mehra (1989)

The algorithm uses the explicit iterative scheme:

$$V_{n+1}(k,\lambda) = \max_{k,n}\left[U(k,k_1,n,\lambda) + \beta\sum_{\lambda'\in\Lambda} p_{\lambda\lambda'} V_n\left(k_1,\lambda'\right)\right] \tag{29}$$

In implementing this procedure, we follow (Danthine, Donaldson & Mehra, 1989). The solution is obtained at a discrete uniform grid of $k$. Specifically, we set the examine the interval $k \in [0.5, 1.5]$ with intervals of 0.0025 implying 401 grid points. To accelerate the numerical maximization of right-hand side in eq.(29), we can decrease the dimensionality of the problem from 2 to1 by tabulating the function $n(k,k_1,\lambda)$ on the two-dimensional grid $k \times k_1$ using the numerical solution of the Euler equation

$$U_3(k,k_1,n,\lambda) = 0 \tag{30}$$

Then this function is substituted into eq.(29). The maximization of

$$U\left(k, k_1, n\left(k, k_1, \lambda\right), \lambda\right) + \beta \sum_{\lambda' \in \Lambda} p_{\lambda\lambda'} V_n\left(k_1, \lambda'\right)$$

for each discrete value of $k$ is performed by choosing the maximal element from the array produced by evaluating the expression for all values of $k_1$. The iterative process defined in eq.(29) is performed until the desired accuracy is achieved.

### 5.1.3 Linear Approximation

We describe here the linearization procedure, because it slightly different from the typical procedure. We use the first order terms of series (28):

$$\begin{aligned} k_1\left(k, \lambda\right) &= k_s + a_1^1\left(k - k_s\right) + a_0^1\left(\lambda - 1\right), \\ n\left(k, \lambda\right) &= n_s + b_1^1\left(k - k_s\right) + b_0^1\left(\lambda - 1\right) \end{aligned} \tag{31}$$

to produce the linear approximation and the following formula for the log-linear approximation:

$$\begin{aligned} \ln k_1\left(k, \lambda\right) &= \ln k_s + a_1^1\left(\ln k - \ln k_s\right) + \frac{a_0^1}{k_s}\ln\lambda, \\ \ln n\left(k, \lambda\right) &= \ln n_s + \frac{k_s}{n_s}b_1^1\left(\ln k - \ln k_s\right) + \frac{b_0^1}{n_s}\ln\lambda \end{aligned} \tag{32}$$

For that purpose we consider the continuous shock modification of system (6) and (7). To make this transition we should replace the sum over discrete states in the expectation operator of the first equation by the integral:

$$\begin{aligned} & U_2\left(k, k_1\left(k, \lambda\right), n\left(k, \lambda\right), \lambda\right) + \\ & \beta \int_{-\infty}^{\infty} \phi\left(\lambda, \lambda'\right) U_1\left[k_1\left(k, \lambda\right), k_1\left(k_1\left(k, \lambda\right), \lambda'\right), n\left(k_1\left(k, \lambda\right), \lambda'\right), \lambda'\right] d\lambda' = 0 \end{aligned} \tag{33}$$

with the conditional probability density function $\phi\left(\lambda, \lambda'\right)$. Assuming that the shocks follow AR(1) process with normally distributed residuals:

$$\lambda_{t+1} = 1 - \rho + \rho\lambda_t + \varepsilon_{t+1},\ \varepsilon_t \tilde{} N\left(0, \sigma\right), \tag{34}$$

we obtain the specific form of function $\phi$:

$$\phi\left(\lambda, \lambda^{'}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\left(\lambda^{'} - \rho\lambda + \rho - 1\right)^{2}}{2\sigma^{2}}\right] \tag{35}$$

The equations for $a^1_{0,1}$ and $b^1_{0,1}$ may be then derived by the differentiation of (33) and (7) w.r.t. $k$ and $\lambda$ at the steady-state point $(k, \lambda) = (k_s, 1)$ with subsequent proceeding to the limit $\sigma \to 0$ (that corresponds to the precision bounds of linear approximation):

$$\begin{aligned} \left(U_{22} + \beta U_{11}\right) a^1_1 + U_{23} b^1_1 + \beta U_{12} \left(a^1_1\right)^2 + \beta U_{13} a^1_1 b^1_1 + U_{12} = 0, \\ U_{23} a^1_1 + U_{33} b^1_1 + U_{13} = 0, \end{aligned} \tag{36}$$

$$\begin{aligned} \left[U_{22} + \beta U_{11} + \beta U_{12}\left(a^1_1 + \rho\right) + \beta U_{13} b^1_1\right] a^1_0 + \left(U_{23} + \beta\rho U_{13}\right) b^1_0 + U_{24} + \beta\rho U_{14} = 0, \\ U_{23} a^1_0 + U_{33} b^1_0 + U_{34} = 0, \end{aligned} \tag{37}$$

where indices denote partial derivatives of function $U\left(k, k_1, n, \lambda\right)$ in the same way as above. The system is quadratic in $a^1_1$. We should pick up the solution with $\left|a^1_1\right| < 1$ corresponding to the contracting mapping. The chosen numeric values of parameters $\rho = 0.95$ and $\sigma = 0.007$ lead to the same statistical properties of time series generated (unconditional correlation between two subsequent shocks $\lambda_t, \lambda_{t+1}$ and standard deviation $E\left(\lambda_t - 1\right)^2$) as those in our discrete state models.

### 5.1.4 Notes on Programming

All algorithms and procedures were programmed using *Mathematica 5.0*; all programs are available from the authors.

**RUGS** As described earlier, the basic one-pass algorithm of RUGS method requires the numerical solution of an algebraic system, depending on the continuous parameter $k$. The corresponding numerical algorithm available in *Mathematica* 5 is the Newton algorithm used in the IDA method of the **NDSolve** function. More detail description of the method can be found in Hindmarsh et. al (2004) and Brenan, Campbell & Petzold (1996).

For our application it is important that the NDSolve function employs an adaptive scheme, i.e. the numeric step of the solution is variable and depends on the required accuracy. The solution itself is returned in the form of a piecewise-polynomial interpolating function of (given by the **InterpolatingFunction** object in Mathematica). Note that these features imply that it is not necessary to create a discrete grid for the capital stock.

**Projection** The numeric solution of nonlinear equation eq.(27) is performed using the **FindRoot** function in *Mathematica*; this uses the Newton algorithm.

**Perturbation** The method requires the sequential calculation of partial derivatives so that a programming language with the capability of symbolic calculations is highly useful. Hence the usage of *Mathematica* is critical for this method.

**Value Function Iteration** The Value Function Iteration algorithm does not require any unique properties of *Mathematica*.

## References

[1] Aruoba, S.B., Fernández-Villaverde, J. and Rubio-Ramírez, J. (2004), "Comparing Solution Methods for Dynamic Equilibrium Economies", Mimeo, University of Pennsylvania.

[2] Barro, R. J., 2005, "Rare Events and the Equity Premium," *National Bureau of Economic Research Working Paper* 11310.

[3] Bloom, N. (2005), "The uncertainty impact of major shocks: firm level estimation and a 9/11 simulation", LSE/Stanford mimeo.

[4] Brenan, K. E., Campbell, S. L., and Petzold, L. R. (1996), Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. SIAM, Philadelphia, Pa.

[5] Boyd, J.P. (2001), *Chebyshev and Fourier Spectral Methods* (Second Revised Edition), Dover Publications, Inc., Mineola, New York, USA.

[6] Christiano, L.J. (1990), "Linear-Quadratic Approximation and Value-Function Iteration: A Comparison," *Journal of Business Economics and Statistics* 8, 99-113.

[7] Danthine, J.-P., Donaldson, J. B., Mehra, R. (1989), "On Some Computational Aspects of Equilibrium Business Cycle Theory", *Journal of Economic Dynamics and Control* 13 449-470.

[8] Hansen, G.D. (1985), "Indivisible Labor and the Business Cycle", *Journal of Monetary Economics* 16, 309-327.

[9] Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R. (2004), Shumaker, and Woodward C. S., *SUNDIALS: Suite of Nonlinear and Differential D. E. /Algebraic Equation Solvers*, LLNL technical report UCRL-JP-200037.

[10] Judd, K. L., and Guu, S-M (1997), "Asymptotic Methods for Aggregate Growth Models", *Journal of Economic Dynamics and Control*, 21, 1025-1042.

[11] Judd, K.L. (1998). *Numerical Methods in Economics.* MIT Press.

[12] Judd, K. L. (1996). "Approximation, Perturbation, and Projection Methods in Economic Analysis",. in *Handbook of Computational Economics*, edited by Hans Amman, David Kendrick, and John Rust, North Holland.

[13] Judd, K.L. & Jin, H.-H.(2002). "Perturbation Methods for General Dynamic Stochastic Models". Mimeo, Hoover Institution.

[14] Kim, J. and Kim, S (2003), "Spurious Welfare Reversals in International Business Cycle Models," *Journal of International Economics,* 60, 471-500.

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

[16] Magill, J.P.M. (1977), "A Local Analysis of N-Sector Capital under Uncertainty," *Journal of Economic Theory* 15, 221-219.

[17] McGrattan, E.R. (1999), "Application of Weighted Residuals Methods to Dynamic Economic Models," in R. Marimon and A. Scott (eds), *Computational Methods for the Study of Dynamic Economies*, Oxford University Press.

[18] Schmitt-Grohe, S. and 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.

[19] Sims, C.A. (2000), "Second Order Accurate Solution of Discrete Time Dynamic Equilibrium Models," Mimeo. Princeton University.

[20] Tesar, L. (1995), "Evaluating the Gains from International Risksharing," *Carnegie-Rochester Conference Series on Public Policy*, 42, 95-143.