Computation of Business Cycle Models: A Comparison of Numerical Methods

Burkhard Heer[a,b,*] and Alfred Maußner[c]

[a] Free University of Bolzano-Bozen, School of Economics and Management, Via Sernesi 1, 39100 Bolzano, Italy, Burkhard.Heer@unibz.it

[b] CESifo, Munich

[c] University of Augsburg, Department of Economics, Universitätsstraße 16, 86159 Augsburg, Germany, alfred.maussner@wiwi.uni-augsburg.de

Revised Version
January 2005

JEL classification: C68, C63, E32

Key Words: Log-Linearization, Perturbation Methods, Projection Methods, Extended Path, Value Function Iteration, Parameterized Expectations, Genetic Search

Abstract:
We compare the numerical methods that are most widely applied in the computation of the standard business cycle model with flexible labor. The numerical techniques imply economically insignificant differences with regard to business cycle summary statistics except in rare cases. Furthermore, these results are robust with regard to the choice of the functional form of the utility function and the model's parameterization. In addition, projection and parameterized expectations methods are found to be more accurate than perturbation methods, even though they are much slower and more difficult to implement.

*Part of this work has been completed while Burkhard Heer was visiting Georgetown University and Stanford University. Burkhard Heer greatfully acknowledges support from the German Science Foundation (DFG) during this period.

1 Introduction

The dominant paradigm of modern business cycle theory is the stochastic growth model with flexible labor. The stochastic growth model, however, is difficult to compute as multiplicative elements such as the production function interact with additive elements such as depreciation or investment. As a consequence, only special cases (with log utility and full depreciation) can be solved analytically. For this reason, the comparison of different computational techniques that approximate the solution numerically is important. Previous work by Taylor and Uhlig (1990) has focused on the study of the stochastic growth model with inelastic labor supply, while Christiano and Fisher (2000) compare different numerical techniques for the solution of the stochastic growth model with binding constraints on nonnegative investment.

In the present paper, we evaluate the numerical techniques that are most widely applied in recent research on non-linear rational expectations general equilibrium models from a different angle. We analyze the properties of standard methods with regard to their accuracy and appropriateness for business cycle research. Importantly for the study of business cycles, we introduce flexible labor supply in the stochastic growth model. Furthermore, we study the sensitivity of the usual set of second moments of the variables that are important for the business cycle researcher, i.e. output, employment, investment, consumption, and wages, with regard to the computational method varying both the functional form of the utility function and the parameterization of the model. In particular, we apply parameter values in the range that are typically observed across countries.

We compare six different computation methods for two different parameter sets. The first set of parameters is chosen with regard to US postwar data. The second set of parameters characterizes the German economy prior to unification. The solution methods are: 1) log-linearization (LL), 2) second order approximation (SO), 3) parameterized expectations (PE), 4) Galerkin projection (GA), 5) extended deterministic path (EP), and 6) value function iteration (VI). For the researcher who is only interested in the business cycle statistics, the log-linearization method is found to be the most convenient and appropriate for the standard business cycle model. Even if the researcher is interested in a more accurate solution, the log-linear solution might provide a good initial value for the parameters of more complex methods. In terms of accuracy, however, we find that methods 3) through 6) clearly dominate methods 1) and 2) that rely upon the local approximation of the solution. Moreover, higher-order perturbation methods do not need to outperform log-linearization in models with flexible labor supply in

general.[1]

The paper is organized as follows. In section 2, the model is presented. In section 3, we briefly review the methods most relevant for the computation of modern business cycle models, and section 4 presents summary statistics for the various methods. Section 5 concludes.

2 The Standard Business Cycle Model

We consider a decentralized economy with households and firms. Households maximize their expected life-time utility

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_t, n_t) \tag{1}$$

with respect to consumption c_0 and labor supply n_0 in period $t = 0$. The time endowment is normalized to one so that $1 - n_t$ denotes leisure in period t. Utility is discounted by $\beta \in (0, 1)$. Instantaneous utility in period t will be chosen among the following functional forms that are commonly applied in business cycle models:

$$u(c_t, n_t) = \begin{cases} \frac{1}{1-\eta}\left[c_t^{1-\eta}(1-n_t)^{\theta(1-\eta)} - 1\right] & \text{I} \\ \frac{1}{1-\eta}\left[\left(c_t - \frac{\theta}{1+\nu} A_t n_t^{1+\nu}\right)^{1-\eta} - 1\right] & \text{II} \\ \ln c_t - \theta n_t & \text{III} \\ \ln c_t + \frac{\theta(1-n_t)^{1-\gamma}}{1-\gamma} & \text{IV} \end{cases} \tag{2}$$

The functions I,III, and IV meet the requirements of King, Plosser, and Rebelo (1988), p.292, that allow for a balanced growth path in the presence of exogenous labor augmenting technical progress A_t.[2] If one uses specification II, a balanced growth path exists only if one is willing to assume that the disutility of work is proportional to A_t.[3] We use the common parameter θ to ensure that the fraction of working hours per worker in total hours available equals the respective empirical magnitude.

[1]This finding is in accordance with the results of a related, but independent study by Aruoba, Fernández-Villaverde and Rubio-Ramírez (2006).

[2]Function I is the standard functional type. Function III models indivisible labor as in Hansen (1985) and is able to explain the fact that total hours and the number of employed workers is much more variable than individual working hours. Function IV, finally, is used by Castañeda, Díaz-Giminénez andRíos-Rull (2004) in their work on heterogeneous-agent economies. For this type of the utility function, working hours vary less with individual productivity and are in better accordance with empirical observations.

[3]This functional form is suggested by Greenwood, Hercowitz, and Huffman (1998) and has the attractive feature that there is no wealth effect on the labor supply decision. Hence, richer households do not supply less labor ceteris paribus.

The households receives income from capital k_t and labor n_t. The budget constraint in period t is given by:

$$k_{t+1} = (1 + r_t)k_t + w_t n_t - c_t, \tag{3}$$

where r_t and w_t denote the real interest rate and the wage rate, respectively.

Firms are of measure one and produce output with labor N_t and capital K_t with constant returns to scale according to:

$$Y_t = Z_t F(K_t, A_t N_t) = Z_t (A_t N_t)^{\alpha} K_t^{1-\alpha}, \tag{4}$$

where A_t denotes the level of labor augmenting technical progress that grows deterministically at the rate $a - 1 \geq 0$:

$$A_{t+1} = a A_t. \tag{5}$$

Total factor productivity Z_t follows the stochastic AR(1)-process:

$$\ln Z_t = \rho \ln Z_{t-1} + \epsilon_t, \tag{6}$$

where ϵ is a serially uncorrelated, normally distributed random variable with mean 0 and variance σ^2.

In a factor market equilibrium, factors are rewarded with their marginal products:

$$r_t = \frac{\partial Z_t F(K_t, N_t)}{\partial K_t} - \delta, \tag{7}$$

$$w_t = \frac{\partial Z_t F(K_t, N_t)}{\partial N_t}. \tag{8}$$

Capital depreciates at rate δ.

In a competitive equilibrium, aggregate variables equal individual variables, $k_t = K_t$, $n_t = N_t$, $c_t = C_t$, and the economy-wide resource constraint is given by $Y_t = K_{t+1} - (1 - \delta)K_t + C_t$. The solution of the model consists of policy functions of the household for consumption $c(k_t, Z_t)$, labor supply $n(k_t, Z_t)$ and next-period capital $k'(k_t, Z_t)$ that cannot be solved analytically in the presence of elastic labor supply (except for specification I, if $\eta \equiv 1$ and $\delta = 1$).

3 Computation of Solutions

The solution of the model satisfies the following set of conditions, where – different from the previous usage – lower case variables are scaled by A_t, i.e., $x_t \equiv X_t/A_t$, except for the shadow price of wealth Λ_t, where $\lambda_t := \Lambda_t A_t^{\eta}$:

$$\frac{\partial u(c_t, N_t)}{\partial c_t} = \lambda_t, \tag{9a}$$

$$\frac{\partial u(c_t, N_t)}{\partial N_t} = \alpha \lambda_t Z_t N_t^{\alpha-1} k_t^{1-\alpha}, \tag{9b}$$

$$a k_{t+1} = Z_t N_t^{\alpha} k_t^{1-\alpha} + (1-\delta) k_t - c_t, \tag{9c}$$

$$\lambda_t = \beta a^{-\eta} E_t \lambda_{t+1} (1 - \delta + (1-\alpha) Z_{t+1} N_{t+1}^{\alpha} k_{t+1}^{-\alpha}) \tag{9d}$$

Due to the recursive structure of the model, the solution can be represented by policy functions $x = g^x(k_t, Z_t)$, $x \in \{c_t, N_t, k_{t+1}\}$, that relate consumption c_t, hours N_t, and the next period capital stock k_{t+1} to the current state (k_t, Z_t) of the system.

We solve this model with six different methods: 1) Log-linearization and 2) second-order approximation, both belonging to the class of perturbation methods that use local information to compute the parameters of g^x, 3) parameterized expectations and 4) Galerkin projection that are well-known examples of projection or weighted residuals methods, 5) deterministic extended path, and 6) value function iteration.[4]

The *log-linear solution method* and *second-order approximation* of the policy function use local information to determine the parameters of the policy functions. Both methods are special cases of perturbation methods (see Judd, 1998, Chapter 13 and 14) that rest on the implicit function theorem and on Taylor's theorem. Let

$$\tilde{x} = g^x(\tilde{k}_t, \tilde{z}_t, \sigma), \quad \tilde{x} \in \{\tilde{k}_{t+1}, \tilde{c}_t, \tilde{N}_t, \tilde{\lambda}_t\}$$

denote the policy function for the natural logarithm of the stock of next-period capital, consumption, working hours, and marginal utility, respectively. The system of first-

[4]Except for the second-order approximation, a detailed description of these methods is provided by Heer/Maußner (2005). The Fortran program that performs all the computations referred to in the text can be downloaded from Alfred Maußner's homepage 'http://www.wiwi.uni-augsburg.de/vwl/maussner/'. It allows the user to supply its own parameter set and has a lot of options for the computation of solutions and test statistics.

order conditions (9) can be written as

$$0 = u_c(g^c(\tilde{k}_t, \tilde{Z}_t, \sigma), g^N(\tilde{k}_t, \tilde{Z}_t, \sigma)) - g^\lambda(\tilde{k}_t, \tilde{Z}_t, \sigma), \tag{10a}$$

$$0 = u_N(g^c(\tilde{k}_t, \tilde{Z}_t, \sigma), g^N(\tilde{k}_t, \tilde{Z}_t, \sigma)) - \alpha e^{g^\lambda(\tilde{k}_t, \tilde{Z}_t, \sigma) + \tilde{Z}_t + (\alpha-1) g^N(\tilde{k}_t, \tilde{Z}_t, \sigma) + (1-\alpha)\tilde{k}_t}, \tag{10b}$$

$$0 = a e^{g^k(\tilde{k}_t, \tilde{Z}_t, \sigma)} - e^{\tilde{Z}_t + \alpha g^N(\tilde{k}_t, \tilde{Z}_t, \sigma) + (1-\alpha)\tilde{k}_t} - (1-\delta) e^{\tilde{k}_t} + e^{g^c(\tilde{k}_t, \tilde{Z}_t, \sigma)}, \tag{10c}$$

$$0 = e^{g^\lambda(\tilde{k}_t, \tilde{Z}_t, \sigma)} - \beta a^{-\eta} E_t e^{g^\lambda(g^k(\tilde{k}_t, \tilde{Z}_t, \sigma), \rho\tilde{Z}_t + \sigma\nu_{t+1}, \sigma)} (1 - \delta + (1-\alpha) e^{w_{t+1}}), \tag{10d}$$

$$w_{t+1} := \rho\tilde{Z}_t + \sigma\nu_{t+1} + \alpha g^N(g^k(\tilde{k}_t, \tilde{Z}_t, \sigma), \rho\tilde{Z}_t + \sigma\nu_{t+1}, \sigma) - \alpha g^k(\tilde{k}_t, \tilde{Z}_t, \sigma), \tag{10e}$$

$$\nu_t \sim N(0, 1). \tag{10f}$$

At $\sigma = 0$, the solution of this system of equations provides the stationary values of the log of capital, consumption, and working hours, $\tilde{k}^*$, $\tilde{c}^*$, and $\tilde{N}^*$, respectively. The coefficients of the log-linear approximation of the policy functions,

$$\tilde{x} = \tilde{x}^* + g^x_k(\tilde{k}_t - \tilde{k}^*) + g^x_Z(\tilde{Z}_t - \tilde{Z}^*) + g^x_\sigma \sigma$$

can be found by differentiating (10a) through (10d) with respect to $\tilde{k}_t$, $\tilde{Z}_t$, and σ at the point $\sigma = 0$. This delivers a system of equations in the twelve unknown coefficients of the four policy functions. As Schmitt-Grohé and Uribe (2004) have shown, g^x_k and g^x_Z are independent of σ and $g^x_\sigma = 0$. Also, it is easy to show that the same coefficients are obtained by first log-linearizing the system (9) at the stationary equilibrium and then solving the resulting stochastic linear system of difference equations. Procedures to solve this system are proposed, among others, by Blanchard and Kahn (1980), Uhlig (1999), and King and Watson (2002). We employ the solution proposed by King and Watson (2002) that rests on the numerically very accurate and stable Schur factorization.

Differentiating the system (10a) twice at $\sigma = 0$ delivers systems of linear equations that can be solved for the coefficients of the quadratic part in the second-order approximation of g^x:

$$\begin{aligned}\tilde{x} = \tilde{x}^* &+ g^x_k(\tilde{k}_t - \tilde{k}^*) + g^x_Z(\tilde{Z}_t - \tilde{Z}^*) + g^x_\sigma \sigma \\ &+ \frac{1}{2} \begin{bmatrix} \tilde{k}_t - \tilde{k}^* & \tilde{Z}_t - \tilde{Z}^* & \sigma \end{bmatrix} \begin{bmatrix} g^x_{kk} & g^x_{kZ} & g^x_{k\sigma} \\ g^x_{Zk} & g^x_{ZZ} & g^x_{Z\sigma} \\ g^x_{\sigma k} & g^x_{\sigma Z} & g^x_{\sigma\sigma} \end{bmatrix} \begin{bmatrix} \tilde{k}_t - \tilde{k}^* \\ \tilde{Z}_t - \tilde{Z}^* \\ \sigma \end{bmatrix}.\end{aligned}$$

Schmitt-Grohé and Uribe (2004) show that even the coefficients g^x_{kk}, g^x_{ZZ}, and $g^x_{kZ} = g^x_{Zk}$ are independent of σ. Furthermore, they prove that $g^x_{k\sigma} = g^x_{\sigma k} = 0$ and $g^x_{Z\sigma} = g^x_{\sigma Z} = 0$. In general it is very cumbersome and prone to failure to derive the second order partial derivatives analytically. Fortunately, there exists software like the MATLAB toolbox

Symbolic Math or Mathematica that is able to execute symbolic differentiation. For our simple model it is not that difficult to employ paper and pencil to find the formulas from which our Fortran program computes the g^x_{ij}.

The *parameterized expectations approach* approximates the rhs of (9d) by a polynomial in $(\ln Z, \ln k)$.[5] We use a simple, complete, exponential polynomial of second degree,[6]

$$\psi(\boldsymbol{\gamma}, \ln Z, \ln k) := \exp(\gamma_1 + \gamma_2 \ln Z + \gamma_3 \ln k + \gamma_4 (\ln z)^2 + \gamma_5 (\ln k)^2 + \gamma_6 \ln z \ln k). \quad (11)$$

The vector of parameters $\boldsymbol{\gamma}$ is determined as solution to a non-linear set of equations. This system depends itself on a long series of points $\{Z_t\}_{t=0}^T$, $\{k_t\}_{t=0}^T$, $\{\lambda_t\}_{t=0}^T$, $\{N_t\}_{t=0}^T$, and $\{c_t\}_{t=0}^T$ obtained from iterations over

$$\lambda_t = \psi(\boldsymbol{\gamma}, \ln Z_t, \ln k_t), \quad (12a)$$

$$\frac{\partial u(c_t, N_t)}{\partial c_t} = \lambda_t, \quad (12b)$$

$$\frac{\partial u(c_t, N_t)}{\partial N_t} = \lambda_t \alpha Z_t N_t^{\alpha-1} k_t^{1-\alpha}, \quad (12c)$$

$$a k_{t+1} = Z_t N_t^{\alpha} k_t^{1-\alpha} + (1-\delta) k_t - c_t, \quad (12d)$$

where Z_t is obtained from (6) using a random number generator that provides pseudo normally distributed innovations ϵ_t. Given λ_t equations (12b) and (12c) can be reduced to an equation that determines N_t given (Z_t, k_t). This equation depends upon the choice of utility function and is given by

$$0 = (1-N_t)^{\theta(1-\eta)/\eta - 1} - (\alpha/\theta)\lambda_t^{1/\eta} Z_t N_t^{\alpha-1} k_t^{1-\alpha}, \quad \text{(I)}$$

$$0 = N_t^{1+\nu-\alpha} - (\alpha/\theta) Z_t k_t^{1-\alpha}, \quad \text{(II)}$$

$$0 = N_t^{1-\alpha} - (\alpha/\theta)\lambda_t Z_t k_t^{1-\alpha}, \quad \text{(III)}$$

$$0 = 1 - (\alpha/\theta)(1-N)^{\gamma} \lambda_t Z_t N_t^{\alpha-1} k_t^{1-\alpha}. \quad \text{(IV)}$$

Given N_t it is easy to solve for c_t. For initial values of the parameters and the simulated time series, we can compute the least squares of the residuals $R(\ln Z_t, \ln k_t) = C_{t+1}^{-\eta} - \psi(\boldsymbol{\gamma}, \ln Z_t, \ln k_t)$. The parameter vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_6)$ is the solution of the following system of non-linear equations:

$$0 = \frac{-2}{T} \sum_{t=0}^{T-1} R(\ln Z_t, \ln k_t) \frac{\partial \psi(\boldsymbol{\gamma})}{\partial \gamma_i}, \quad i = 1, 2 \ldots, 6. \quad (14)$$

[5] The method of parameterized expectations can be interpreted as a special case of the more general class of projections methods as pointed out by Judd (1996). Furthermore, he emphasizes that the approach was originally developed by Williams and Wright (1982,1984,1991).

[6] See, for example, den Haan and Marcet (1994).

The crucial step in applying this algorithm is to find acceptable starting values for the non-linear equations solver. One way to tackle this problem is to obtain linear approximations of the policy functions for consumption and labor supply, to use these to compute time series for λ_t, k_t, and Z_t, and to regress λ_t on $\psi(\ln k_t, \ln Z_t)$ by way of non-linear least squares.[7] If this works, the parameterized expectations algorithm is a fast way to find the final solution. However, in most cases it did not work. For the German parameter set we were not able to find admissible starting values for all four specifications of the utility function. For the US parameter set utility function II posed the same problem. In these cases, we used a genetic search algorithm.[8] To reduce computation time this algorithm operated over short time series of 5,000 points. When a solution was found, we used it as a starting value for a system of non-linear equations based on the final number of periods $T = 100,000$. This large number of points guarantees a good sample of the underlying stationary distribution implicitly defined by the system of stochastic difference equations (9) so that the final solution is very accurate.

As in the parameterized expectations approach, the *Galerkin projection method* rests on the approximation of the rhs of (9d). Contrary from the previous method, however, we use a third degree product base, Chebyshev exponential polynomial in $(\ln Z, \ln k)$ as the approximating function:

$$\psi(\ln k_t, \ln Z_t) := \exp\left[\sum_{i=0}^{p_1}\sum_{j=0}^{p_2}\gamma_{ij}T_i\left(X(\ln Z_t)\right)T_j\left(X(\ln k_t)\right)\right],$$

where $T_i(\cdot)$ is the i-th degree Chebyshev polynomial and $X(x)$ denotes the linear transformation of $[\underline{x}, \overline{x}]$, $x \in \{\ln Z, \ln k\}$ to $[-1, 1]$, the domain of Chebyshev polynomials. Given $\psi(\cdot)$ and $(\ln k, \ln Z)$ we are able to compute for each innovation ϵ the rhs of (9d): First, we solve (12) for $k' := k_{t+1}$. Letting $z' := \rho \ln Z_t + \epsilon$ and $N' := N_{t+1}$ we use $(\ln k', z')$ and solve (12) again to get

$$g(\epsilon, \ln Z, \ln k) := \psi(\ln k', z')(1 - \delta + (1 - \alpha)e^{z'}(N')^{\alpha}(k')^{-\alpha}).$$

[7] Since the error term in $\lambda_t - \psi(\ln k_t, \ln Z_t)$ is additive it is inappropriate to regress $\ln \lambda_t$ on $\ln k_t$ and $\ln Z_t$ by means of ordinary least squares.

[8] In particular, we applied two different specifications of the genetic search algorithm. In our first specification, we follow Duffy and McNelis (2001). Yet, different from their paper, our fitness criterium is the minimal absolute value of the rhs of (14). Second, we use a different selection scheme and employ a larger set of cross-over operators than Duffy and McNelis (2001). In particular, we use stochastic universal sampling as in Mitchell (1996). The genetic search algorithms are described in more detail in Heer and Maußner (2005). None of the two algorithms is found to dominate the other in terms of speed across all calibrations and parameterizations in our model.

The residual function $R(\boldsymbol{\gamma}, \ln k, \ln Z)$ that is obtained from (9d) by using $\psi(\cdot)$ instead of the true but unknown solution is now given by

$$R(\boldsymbol{\gamma}, \ln k, \ln Z) := \psi(\ln k, \ln Z) - \beta a^{-1} \int_{\ln \underline{Z}}^{\ln \overline{Z}} g(\epsilon, \ln k, \ln Z)(2\pi\sigma^2)^{-1/2} e^{-1/2\sigma^2} \epsilon^2 d\epsilon.$$

The Galerkin projection method chooses $\boldsymbol{\gamma}$ to solve[9]

$$\int_{\ln \underline{k}}^{\ln \overline{k}} \int_{\ln \underline{Z}}^{\ln \overline{Z}} R(\boldsymbol{\gamma}, \ln k, \ln Z) T_i(X(\ln Z)) T_j(X(\ln k)) d\ln k \; d\ln Z = 0,$$
$$i = 1, 2, \ldots, p_1, \;\; j = 1, 2, \ldots, p_2. \tag{15}$$

We use Gauss-Chebyshev quadrature with 50 nodes in each dimension to compute this double integral. The critical step is the choice of the interval $[\ln \underline{k}, \ln \overline{k}]$ for the capital stock so that the algorithm always stays within this interval. We solve this problem in the following way that, to the best of our knowledge, has not been emphasized in the previous literature on projection methods: We simply use a larger interval that embeds the latter but integrate over the smaller only. More exactly, the conditional expectation is approximated over the interval $[\ln k_{min}, \ln k_{max}]$, while the Galerkin integral in (15) is computed over $[\ln \underline{k}, \ln \overline{k}] \subset [\ln k_{min}, \ln k_{max}]$. While our program does not find a solution for $\ln \underline{k} = k_{min}$ and $\ln \overline{k} = \ln k_{max}$, it converges for $\ln \underline{k} = 1.3 \cdot k_{min}$ and $\ln \overline{k} = \ln k_{max}/1.3$. The basic reason for this behavior of the algorithm is that for initial values of $\boldsymbol{\gamma}$ in the parameterized function, $\ln k_{t+1}$ might happen to fall outside the interval $[\ln k_{min}, \ln k_{max}]$ if we choose $[\ln \underline{k}, \ln \overline{k}]$ too wide. In this case, however, we get highly inaccurate solutions for the policy functions. Note, that the accuracy of the solution depends on p_1 and p_2 for given integration bounds in (15) and not on the size of $[\ln k_{min}, \ln k_{max}]$. We found that $p_1 = p_2 = 3$ provides a very good approximation.

As in the case of the parameterized expectations method we use the linear approximations of the solution for λ_t to initialize the non-linear equations solver. This worked in all but two cases. For utility functions I and III and the German parameter set the linear policy function did not provide a feasible initial parameter vector. Fortunately, in both cases the solution from utility function IV provided admissible starting values.

The *extended deterministic path method* assumes that after a shock in period t no further shock occurs and computes the dynamics for the next T periods. Therefore, it has to solves a set of $2T - 1$ equations in the unknowns $N_{t+s}, k_{t+s+1}, s = 0, 2, T - 1$ obtained from (9a) to (9d) assuming that k_{t+T} is equal to the respective stationary

[9] For a more detailed description of the projection methods, see Judd (1992, 1998) and McGrattan (1999).

solution of the deterministic counterpart of the model. From this solution only N_t and k_{t+1} are retained. Then another shock is drawn and the respective systems of equations is solved for N_{t+1} and k_{t+2} and so forth. The accuracy of the solution depends upon T. We found that $T = 150$ gives a very accurate solution, yet at the cost of long computation time.

The *value function iteration method* iterates on:

$$v^{s+1}(k_t, Z_t) := \max_{k_{t+1}, N_t} u(Z_t N_t^{\alpha} k_t^{1-\alpha} + (1-\delta)k_t - ak_{t+1}, N_t) + \beta a^{1-\eta} E_t v^s(k_{t+1}, Z_{t+1}), \quad (16)$$

where $v^s(\cdot)$ denotes the value function at iteration s. We use a discrete version of our model to perform these iterations: We approximate the continuous AR(1)-process (6) by a finite Markov chain $\mathscr{Z} = \{z_1, z_2, \ldots, z_m\}$ of m elements and transition matrix $P = (p_{ij})$, where p_{ij} denotes the probability to move from state z_i to state z_j (see Tauchen, 1986). The difference between $\ln z_m$ and $\ln z_1$ is μ-times the size of the unconditional standard deviation of (6). We choose μ so that even in simulations with a large number of realizations the productivity shock remains in $[\ln z_1, \ln z_m]$. The capital stock can take values from a grid $\mathscr{K} = \{k_1, k_2, \ldots, k_n\}$ of n elements. The upper (lower) bound k_n (k_1) equals the value that the deterministic counterpart of the model would approach if Z_t would equal z_m (z_1) for all $t = 0, 1, \ldots$. Given the grid $\mathscr{G} = \mathscr{K} \times \mathscr{Z}$, the value function is a $n \times m$-matrix $V = (v_{ij})$, where v_{ij} is the maximum expected life-time utility obtained from a sequence of optimal choices starting at $(k_i, z_j) \in \mathscr{G}$ at time $t = 0$. We determine the elements of this matrix by iterating over the discrete counterpart of (16):

$$v_{ij}^{s+1} := \max_{k_h \in \mathscr{K}, N} u(z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - ak_h, N) + \beta a^{1-\eta} \sum_{l=1}^{m} p_{jl} v_{hl}^s. \quad (17)$$

As initial elements of V we use the value obtained from the balanced growth path of a model with productivity level $z_j \in \mathscr{Z}$ for all $t = 0, 1, \ldots$.

Given a triple (z_j, k_i, k_h) we must solve the first order condition for optimal labor supply. Depending on the type of the utility function I-IV, this requires the solution of the following (implicit) equation:

$$0 = z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - ak_h - (\alpha/\theta)(1-N) z_j N^{\alpha-1} k_i^{1-\alpha}, \quad \text{(I')}$$

$$0 = N^{1+\nu-\alpha} - (\alpha/\theta) z_j k_i^{1-\alpha}, \quad \text{(II')}$$

$$0 = z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - ak_h - (\alpha/\theta) z_j N^{\alpha-1} k_i^{1-\alpha}, \quad \text{(III')}$$

$$0 = z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - ak_h - (\alpha/\theta)(1-N)^{\gamma} z_j N^{\alpha-1} k_i^{1-\alpha}. \quad \text{(IV')}$$

It can be shown that each of these equations has a unique solution $N \in [0,1]$ given reasonable values of (k_i, k_h).

The approximate solution of (17) is a $n \times m$ matrix H with the typical element h_{ij} being a pointer to the index h of the capital stock $k_h \in \mathscr{K}$ that is optimal given the current capital stock $k_i \in \mathscr{K}$ and the current productivity level $z_j \in \mathscr{Z}$. Simulations of the model use bilinear interpolation over $\mathscr{G}$ and the associated policy function H to preserve the continuous nature of the productivity shock (6). A more detailed description of our optimization procedure together with a justification of our approach vis-à-vis others is presented in the Appendix.[10]

Obviously, the accuracy of the solution as well as computation time increase with both n and m. With respect to the second moments of simulated time series we found that $n = 5000$ and $m = 9$ are a good compromise between speed and accuracy. In terms of Euler equation residuals (see below) very accurate solutions require $n = 20000$ and $m = 19$.

4 Results

We evaluate the methods with regard to 1) computation time 2) the usual set of second moments computed from simulated, HP-filtered time series of employment, output, investment, consumption, 3) the risk-free rate of return, and 4) accuracy. We also consider the risk-free rate of return in view of the results by Christiano and Fisher (2000) who find that asset price statistics seem to be more sensitive to the accuracy of the solution. Accuracy is measured using the residual of the Euler equation (9d).

As we are aiming to assess the suitability of the different methods for business cycle models more generally, we analyze two different sets of calibration parameters. The first set reflects parameters commonly applied in the business cycle study of the postwar US economy. The second set of parameters is calibrated with the help of German postwar data prior to unification in 1989.[11] Importantly, these two economies are characterized by different institutional settings. In particular, the German capital market is less competitive as many banks are state-owned or subsidized by the state. As a consequence, capital depreciates less rapidly in Germany as capital utilization is

[10] Our method exploits the monotonicity of the policy function and the concavity of the value function. We find this algorithm to be much more accurate for given computation time than Howard's algorithm or a variant that uses interpolation between grid points.

[11] A detailed description of this calibration is provided in Chapter 1 of Heer and Maußner (2005).

lower. Furthermore, capital's share in output, $1-\alpha$, is lower in Germany (0.27) than in the US (0.36). One possible reason may be the presence of unions. Secondly, labor markets are more rigid in Germany and the social security system is more generous. As a consequence, average labor supply is lower in Germany as well.

Table 1
Model Calibration

German Calibration		US Calibration	
Production	Preferences	Production	Preferences
a=1.005	β=0.994	a=1.0055	β=0.99
α=0.73	η=1.00	α=0.64	η=1.00
δ=0.011	ν=5.00	δ=0.025	ν=3.33
ρ=0.90	γ=33.5	ρ=0.95	γ=7.00
σ=0.072	N=0.13	σ=0.0072	N=0.33

For the US economy, we use the set of parameters displayed in Table 1. Except for the rate of per capita output growth they are in accordance with Hansen (1985). The average quarterly growth rate of the US economy exceeds the one of the German economy, $a' = 1.0050$, and amounts to $a = 1.0055$ during 1960-2002 on average. The estimates of the Frisch intertemporal labor supply elasticity $\eta_{n,w}$ implied by microeconometric studies and the implied values of γ and ν vary considerably. MaCurdy (1981) and Altonji (1986) both use PSID data in order to estimate values of 0.23 and 0.28, respectively, while Killingsworth (1983) finds an US labor supply elasticity equal to $\eta_{n,w} = 0.4$.[12] We will use the conservative estimate $\eta_{n,w} = 0.3$ and, accordingly, apply the values $\nu = 3.33$ and $\gamma = 7.0$ in utilities III and IV, respectively.[13] For Germany, we use the same set of parameters as in Heer and Maußner (2005).[14] In addition, we use the value $\eta_{n,w} = 0.2$ following Heer and Trede (2003) implying $\nu = 5.0$ and $\gamma = 33.5$.

[12]Domeij and Floden (2006) argue that these estimates are biased downward due to the omission of borrowing constraints.

[13]Greenwood et al. (1988) even apply a value $\eta_{n,w} = 1.7$ corresponding to $\nu = 0.6$ in their study, while Castañeda et al. (2004) use $\gamma = 5.5$.

[14]In particular, we did not find any compelling evidence that the intertemporal elasticity of substitution is different between the US and Germany.

Table 2
Computation Time

Method	Time
Log-linear approximation (LL)	0.42 s
Second order approximation (SO)	0.52 s
Parameterized expectations (PE)	
- initial value from LL	1 m 21.22 s
- two steps with search (UII, US)	1 m 10.32 s
- two steps with search (UII,GE)	23 m 11.33 s
Galerkin projection (GA)	
- initial value from LL	3 44.36 s
- initial value from utility IV (UI,GE)	2 m 3.86 s
Extended deterministic path (EP)	2 h 27 m 0.86 s
Value function iteration (VI)	
- $n = 5000, m = 9$	21 m 50.62 s
- $n = 20000, m = 9$	2 h 46 m 40.58 s
- $n = 20000, m = 19$	4 h 40 m 46.56 s

Notes:
If not mentioned otherwise, the results are based on the solution for utility function I and the US parameter set. The program run on a 3.2 Gigahertz Pentium IV personal computer. The program is written in Fortran 95 and compiled with the Compaq Digital Fortran developer environment. Computation time comprises the time needed to compute the solution and to compute the summary statistics from 500 simulations. The stochastic search routine rests on a population of 50 candidate solution vectors and iterates over 100 generations. h=hours, m=minutes, s=seconds

Computation Time. The computation time of the solution for the six methods is presented in Table 2. The algorithms can be ordered with respect to running time as follows: 1) Log-linear approximation is by way the fastest method. Using a 3.2 Gigahertz Pentium IV personal computer it takes less than a second to compute the policy function and to run 500 simulations over 60 periods each. 2) If one disregards the time needed to derive the analytical expressions for the second derivatives by way of paper and pencil, the second-order approximation is almost as fast as log-linearization, requiring 0.52 seconds to perform the same task. The further ranks depend on our measurement of computation time and our strategy to find acceptable starting values for the non-linear equations solver. If the algorithm converges using the initial val-

ues obtained from the log-linear solution 3) the parameterized expectations approach (PE) is much faster than 4) the Galerkin projection method (GA) and both need less time than 5) the value function iteration method (VI) and 6) the extended path (EP) method. The PE needs less than one and a half minute whereas the GA consumes more than three and a half minutes of computer time. The reason is that the GA must evaluate a sum over 2500 elements (we use 50 nodes for Z and 50 nodes for k to compute the respective double integral), where each summand (namely the approximation of the conditional expectation) is itself a sum over 100 elements. In the case of the GA it was not necessary to use stochastic search. The search for acceptable starting values for the non-linear equations solver can considerably increase the computation time of the PE solution, namely up to about 24 minutes.[15] The ranking between value function iteration and the extended path method depends upon the desired accuracy. With $n = 5000$ and $m = 9$, VI is less accurate but quite faster than EP. In the case of the German parameter set VI is about as accurate as EP for $n = 20,000$ and $m = 9$. In this case, VI is a bit faster than EP (2 hours 23 minutes versus 2 hours 35 minutes). To obtain about the same accuracy in the case of the US parameter set we had to use $n = 20,000$ and $m = 19$ so that EP clearly outperforms VI. Finally, we like to mention that the solution in the case of utility function II, where an analytical solution for N given k and Z is available, requires substantially less time. For instance, the extended path method finds the solution in about 41 minutes, and value function iteration requires 2 hours and 26 minutes.

Summary Statistics. Our estimation results for the US calibration are displayed in Tables 3-6 for the utility functions I-IV, respectively. Consider Table 3. The first column presents the variable, the next three columns display the standard deviation of the variable, the variable's correlation with output, and its autocorrelation, respectively. As is obvious from the inspection of the table, the minimum and maximum values obtained in the computation with either methods are close to the mean values. Among the 40 measures of volatility there are three instances where the spread between the minimum and the maximum value exceeds the average standard deviation of the respective statistic. All three cases pertain to the US parameter set: consumption for utility II, and investment and hours for utility III.

[15]Computation time can exceed one hour if the search process is applied to the the final number of periods $T = 100,000$. We, however, used a two step procedure: find admissible starting values for a small number of periods $T = 5000$ and use these to compute the solution for the final number of periods $T = 100,000$.

Table 3

Summary Statistics: Utility I

Variable	s_x			s_{xy}			r_x		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
					US Calibration				
Output	1.24	1.26	1.27	1.00	1.00	1.00	0.66	0.66	0.67
Standard Dev.	0.21	0.22	0.23	0.00	0.00	0.00	0.09	0.10	0.11
Investment	3.61	3.66	3.69	0.99	0.99	0.99	0.64	0.65	0.65
Standard Dev.	0.59	0.62	0.67	0.00	0.00	0.00	0.09	0.10	0.11
Consumption	0.43	0.43	0.43	0.90	0.91	0.91	0.75	0.76	0.76
Standard Dev.	0.09	0.09	0.09	0.02	0.02	0.03	0.08	0.09	0.09
Hours	0.59	0.60	0.60	0.98	0.98	0.98	0.64	0.64	0.65
Standard Dev.	0.10	0.10	0.11	0.01	0.01	0.01	0.10	0.10	0.11
Real Wage	0.68	0.69	0.69	0.98	0.98	0.98	0.69	0.69	0.70
Standard Dev.	0.12	0.12	0.13	0.01	0.01	0.01	0.09	0.10	0.10
					German Calibration				
Output	1.91	1.92	1.93	1.00	1.00	1.00	0.62	0.65	0.72
Standard Dev.	0.22	0.30	0.35	0.00	0.00	0.00	0.04	0.09	0.11
Investment	8.49	8.75	8.99	0.99	0.99	1.00	0.62	0.64	0.71
Standard Dev.	0.96	1.41	1.71	0.00	0.00	0.00	0.03	0.09	0.12
Consumption	0.33	0.34	0.39	0.81	0.82	0.84	0.76	0.78	0.82
Standard Dev.	0.07	0.07	0.08	0.02	0.04	0.05	0.05	0.08	0.09
Hours	1.39	1.43	1.46	0.99	0.99	0.99	0.62	0.64	0.72
Standard Dev.	0.15	0.22	0.26	0.00	0.00	0.00	0.03	0.09	0.12
Real Wage	0.51	0.52	0.57	0.94	0.95	0.95	0.68	0.70	0.77
Standard Dev.	0.08	0.09	0.10	0.01	0.02	0.02	0.05	0.09	0.11

Notes:

s_x is the standard deviation of variable x listed in the first column. s_{xy} is the correlation of x with output. s_x denotes the first order autocorrelation of x. All moments are averages over 500 simulations. The length of the respective time series was 60 periods in each simulation. All time series are HP-filtered with weight $\lambda = 1600$. The columns labeled 'Min', 'Mean', and 'Max' display the minimum, the mean, and the maximum from the five different methods, respectively. The rows labeled 'Std. Dev.' give the standard deviation of the respective moment, computed from the 500 simulated time series.

Table 4

Summary Statistics: Utility II

Variable	s_x			s_{xy}			r_x		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
					US Calibration				
Output	1.03	1.04	1.05	1.00	1.00	1.00	0.66	0.67	0.67
Standard Dev.	0.17	0.18	0.18	0.00	0.00	0.00	0.10	0.10	0.10
Investment	2.29	2.53	2.62	0.99	0.99	0.99	0.65	0.66	0.66
Standard Dev.	0.38	0.43	0.45	0.00	0.00	0.00	0.10	0.10	0.11
Consumption	0.48	0.50	0.58	0.97	0.98	0.99	0.68	0.70	0.71
Standard Dev.	0.09	0.09	0.10	0.00	0.01	0.01	0.09	0.10	0.10
Hours	0.24	0.24	0.24	1.00	1.00	1.00	0.66	0.67	0.67
Standard Dev.	0.04	0.04	0.04	0.00	0.00	0.00	0.10	0.10	0.10
Real Wage	0.79	0.80	0.81	1.00	1.00	1.00	0.66	0.67	0.67
Standard Dev.	0.13	0.14	0.14	0.00	0.00	0.00	0.10	0.10	0.10
					German Calibration				
Output	0.98	0.98	0.99	1.00	1.00	1.00	0.63	0.63	0.64
Standard Dev.	0.15	0.16	0.17	0.00	0.00	0.00	0.10	0.11	0.11
Investment	3.77	3.83	3.88	1.00	1.00	1.00	0.62	0.63	0.64
Standard Dev.	0.60	0.62	0.64	0.00	0.00	0.00	0.10	0.11	0.11
Consumption	0.30	0.31	0.33	0.96	0.96	0.97	0.66	0.67	0.68
Standard Dev.	0.05	0.05	0.06	0.01	0.01	0.01	0.10	0.10	0.11
Hours	0.16	0.16	0.17	1.00	1.00	1.00	0.63	0.63	0.64
Standard Dev.	0.03	0.03	0.03	0.00	0.00	0.00	0.10	0.11	0.11
Real Wage	0.82	0.82	0.83	1.00	1.00	1.00	0.63	0.63	0.64
Standard Dev.	0.13	0.13	0.14	0.00	0.00	0.00	0.10	0.11	0.11

Notes: See Table 3.

Table 5

Summary Statistics: Utility III

Variable	s_x			s_{xy}			r_x		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
					US Calibration				
Output	1.46	1.61	1.66	1.00	1.00	1.00	0.65	0.66	0.66
Standard Dev.	0.26	0.27	0.28	0.00	0.00	0.00	0.10	0.10	0.10
Investment	3.89	4.71	4.96	0.99	0.99	0.99	0.64	0.64	0.64
Standard Dev.	0.68	0.79	0.83	0.00	0.00	0.00	0.10	0.10	0.11
Consumption	0.52	0.54	0.61	0.88	0.90	0.95	0.72	0.76	0.77
Standard Dev.	0.11	0.11	0.12	0.01	0.03	0.03	0.09	0.09	0.09
Hours	0.90	1.15	1.23	0.98	0.98	0.98	0.64	0.64	0.64
Standard Dev.	0.16	0.19	0.20	0.01	0.01	0.01	0.10	0.10	0.11
Real Wage	0.52	0.54	0.61	0.88	0.90	0.95	0.72	0.76	0.77
Standard Dev.	0.11	0.11	0.12	0.01	0.03	0.03	0.09	0.09	0.09
					German Calibration				
Output	2.29	2.37	2.40	1.00	1.00	1.00	0.63	0.63	0.63
Standard Dev.	0.37	0.40	0.42	0.00	0.00	0.00	0.10	0.10	0.11
Investment	10.56	11.02	11.28	0.99	0.99	1.00	0.62	0.62	0.63
Standard Dev.	1.82	1.93	2.13	0.00	0.00	0.00	0.10	0.11	0.11
Consumption	0.39	0.40	0.42	0.80	0.81	0.85	0.75	0.77	0.78
Standard Dev.	0.08	0.09	0.09	0.04	0.04	0.05	0.08	0.08	0.09
Hours	1.95	2.07	2.10	0.99	0.99	0.99	0.62	0.62	0.63
Standard Dev.	0.31	0.34	0.36	0.00	0.00	0.00	0.10	0.11	0.11
Real Wage	0.39	0.40	0.42	0.80	0.81	0.85	0.75	0.77	0.78
Standard Dev.	0.08	0.09	0.09	0.04	0.04	0.05	0.08	0.08	0.09

Notes: See Table 3.

Table 6

Summary Statistics: Utility IV

Variable	s_x			s_{xy}			r_x		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
					US Calibration				
Output	0.96	0.97	0.98	1.00	1.00	1.00	0.66	0.66	0.67
Standard Dev.	0.16	0.17	0.17	0.00	0.00	0.00	0.09	0.10	0.11
Investment	2.72	2.75	2.77	0.99	0.99	0.99	0.64	0.65	0.65
Standard Dev.	0.46	0.47	0.48	0.00	0.00	0.00	0.10	0.10	0.11
Consumption	0.35	0.35	0.36	0.92	0.92	0.93	0.74	0.75	0.75
Standard Dev.	0.07	0.07	0.07	0.02	0.02	0.02	0.08	0.09	0.10
Hours	0.15	0.15	0.15	0.98	0.98	0.98	0.64	0.64	0.65
Standard Dev.	0.02	0.03	0.03	0.01	0.01	0.01	0.10	0.10	0.11
Real Wage	0.82	0.83	0.83	1.00	1.00	1.00	0.66	0.67	0.67
Standard Dev.	0.14	0.15	0.15	0.00	0.00	0.00	0.09	0.10	0.11
					German Calibration				
Output	0.96	0.97	0.97	1.00	1.00	1.00	0.63	0.63	0.64
Standard Dev.	0.15	0.16	0.17	0.00	0.00	0.00	0.10	0.10	0.11
Investment	4.14	4.23	4.27	0.99	0.99	1.00	0.62	0.63	0.63
Standard Dev.	0.68	0.71	0.75	0.00	0.00	0.00	0.10	0.10	0.11
Consumption	0.20	0.21	0.23	0.86	0.87	0.90	0.72	0.74	0.75
Standard Dev.	0.04	0.04	0.04	0.03	0.03	0.04	0.09	0.09	0.09
Hours	0.13	0.13	0.13	0.99	0.99	0.99	0.62	0.63	0.63
Standard Dev.	0.02	0.02	0.02	0.00	0.00	0.00	0.10	0.10	0.11
Real Wage	0.83	0.84	0.84	1.00	1.00	1.00	0.63	0.64	0.64
Standard Dev.	0.13	0.14	0.15	0.00	0.00	0.00	0.10	0.10	0.11

Notes: See Table 3.

Notice further that the moments are very much alike for the four different utility functions. If you consider the mean correlation of investment I, consumption C, employment N, and wages W with output Y, displayed in the sixth column of Tables 3-6, the divergence is small. Similarly, the mean autocorrelations of Y, I, C, N, and W, that are displayed in column 9 in each table are hardly discernable. There are only two marked differences between these four parameterizations: 1) the volatility of output (and, similarly, investment) is more pronounced for utilities I and III, and, 2) employment is much more volatile relative to both output and wages in cases I and III as well. Consequently, utility functions I and III are often applied in business cycle research in order to generate the empirically observed (rather high) volatility of output and employment, while utilities II and IV are often applied in heterogenous-agent models in order to replicate the (rather low) dispersion of hours worked among the workers. The accuracy of the six different computational methods for the German parameterization does not significantly differ from the one for the US calibration as is obvious from comparison of the lower half with the upper half of Tables 3-6, respectively.

In summary, all our qualitative results with regard to the computation also hold in the model calibrated for the German economy. With regard to its business cycle features, the two economies are also very much alike with one exception. As can be seen from Table 3 the volatility of output and investment (consumption) is higher (lower) in Germany than the one in the US. As Germans value leisure more than US households (and hence work less in steady state), the relative weight $1/\theta$ of consumption in utility is smaller than in the US for the functional form I of utility. As a consequence, German households have to vary their intertemporal labor supply to a larger extent than US households in order to smooth intertemporal utility and, for this reason, the volatility of hours worked relative to consumption is higher in Germany than in the US. Higher volatility of labor also results in higher volatility of investment and output in Germany.

The Risk-Free Rate of Return Table 7 provides a measure of the risk-free rate of return, $(\lambda_t/(\beta a^{-\eta} E_t \lambda_{t+1})) - 1$. The numbers displayed are percentage differences between the averages from a time series of 100,000 points and the stationary rate $(a^\eta/\beta) - 1$. Since it is very time consuming to compute such a long time series with the extended path method (it required about 2 days), the entries in the column labeled EP are from a shorter series of 10,000 points. In all cases, the numbers are very close to $a^\eta/\beta - 1$. The maximum absolute deviation from this stationary rate is 0.35 percent and occurs in the case of utility function III and the extended path method (German calibration). The observation is probably due to the much shorter time series. The

Table 7

Risk Free Rate of Return

Utility function	EP	GA	LL	PE	SO	VI
			US Calibration			
Utility I	0.099	−0.003	−0.088	0.017	−0.055	−0.067
Utility II	−0.169	−0.048	0.068	−0.103	−0.018	0.026
Utility III	0.051	−0.036	−0.009	−0.032	0.007	−0.077
Utility IV	0.384	−0.036	0.064	−0.032	−0.059	0.079
			German Calibration			
Utility I	0.006	−0.081	0.006	0.065	−0.050	−0.002
Utility II	−0.155	−0.063	−0.073	−0.009	0.061	0.016
Utility III	−0.352	−0.072	−0.043	0.015	0.019	0.128
Utility IV	0.000	0.031	−0.048	0.003	−0.026	0.009

Notes:

EP: extended path, GA: Galerkin projection, LL: log-linear approximation, PE: parameterized expectations SO: second order approximation, VI: value function iteration.

Except for the extended path method the risk free rate of return has been computed as the average of $[\lambda_t/(\beta a^{-\eta} E_t \lambda_{t+1}) - 1$ from a time series with 100,000 points. The entries in the column EP are averages of 10,000 points. The conditional expectation in the definition of the risk free rate is computed using Gauss-Hermite quadrature with four points. The risk free rate of return in the stationary equilibrium is given by $a^\eta/\beta - 1 = 0.0156566$ for the US calibration and by $a^\eta/\beta - 1 = 0.0110664$ for the German calibration.

maximum absolute deviation in columns three to seven is about 0.13 percent and is associated with value function iteration in the case of utility function III. Thus, in general, the variations of the risk-free rate of return over the methods considered are negligible.

Accuracy. The residual e of the Euler equation for capital is computed as $e = (\tilde{c}/c) - 1$, where c is consumption. $\tilde{c}$ is the amount of consumption that is necessary to equate the lhs of equation (9d) to our approximation of $\beta a^{-\eta} E_t u_c(c', 1 - n')(1 - \delta + (1 - \alpha) Z'(N')^\alpha (k')^{-\alpha})$, where the prime denotes next period values of consumption, hours, and capital, respectively. We use four-point Gauss-Hermite quadrature to compute the conditional expectation on the rhs of equation (9d) given the policy function delivered by the respective solution method. We compute the residuals for 400 equally spaced pairs of $(k, z) \in [0.8k^*, 1.2k^*] \times [0.95, 1.05]$. We choose this subset since it is the domain of simulated time series, even for very long ones with 100,000 points.

Table 8 displays the maximum absolute value of the 400 residuals. The methods that

Table 8

Euler Equation Residuals

Utility function	EP	GA	LL	PE	SO	VI
			US Calibration			
Utility I	0.0005832	0.0005849	0.0058276	0.0006070	0.0061760	0.0003779
Utility II	0.0002905	0.0002905	0.0100118	0.0003896	0.0101356	0.0007192
Utility III	0.0007782	0.0007795	0.0056415	0.0006590	0.0057423	0.0002345
Utility IV	0.0004550	0.0004556	0.0064823	0.0005645	0.0067618	0.0006084
			German Calibration			
Utility I	0.0004058	0.0004046	0.0051352	0.0002720	0.0038616	0.0005043
Utility II	0.0004853	0.0001545	0.0038995	0.0002083	0.0040368	0.0008927
Utility III	0.0005107	0.0005069	0.0055942	0.0004658	0.0039372	0.0004791
Utility IV	0.0002191	0.0002179	0.0028265	0.0004944	0.0023303	0.0008325

Notes:

EP: extended path, GA: Galerkin projection, LL: log-linear approximation, PE: parameterized expectations SO: second order approximation, VI: value function iteration.

Each entry in this table is the maximum absolute value of the Euler equation residuals computed over an equally spaced grid of 400 knots in the interval $[0.8k^*, 1.2k^*] \times [0.95, 1.05]$. Following Christiano and Fisher (2000) we computed the Euler residual as $\tilde{c}/c - 1$, where c is optimal consumption given the policy function and $\tilde{c}$ is the amount of consumption that is necessary to equate the lhs of the Euler equation for capital to its rhs. The conditional expectation on the rhs of the Euler equation for capital is computed using Gauss-Hermite quadrature with four points. In the case of the US parameter set we used $n = 20,000$ and $m = 19$ to compute the VI solution, whereas in the case of the German parameter set $n = 20,000$ and $m = 9$ provided about the same accuracy as the EP, GA, and PE methods.

use only local information to determine the parameters of the policy function, namely first- and second-order approximation, are the least accurate ones. Remarkably, there are only three instances where second-order approximation is slightly more accurate than first-order approximation, namely in the case of the German parameter set for utility functions I, III, and IV. The Euler residuals associated with the remaining four methods are about 10 times more accurate. In the case of value function iteration, however, this accuracy requires a grid of $n = 20,000$ points for the stock of capital and $m = 19$ points for the level of technology. Interestingly, we obtain the same size of the residual for a more coarse grid ($m = 9$) over $\mathscr{Z}$ in the case of the German parameter set.[16] If we increase m to 19 points in this case, the maximum Euler equation residual

[16]The VI method with a grid of $n = 5000$ and $m = 9$ delivers Euler equation residuals of about the sane size as in the case of the log-linear approximation. These results, however, are not displayed in Table 8.

further declines from 0.0005043 to 0.0004046, yet at the cost of an increase in running time from about 2 h 22 m to 7h 45 m. Given acceptable starting values both Galerkin projection and parameterized expectations deliver results of the same precision within a few minutes. Note also the very good results obtained from the extended path method. There is, however, no method that dominates the others in terms of accuracy. For instance, value function iteration delivers the smallest Euler residual in the case of the US parameter set for utility functions I and III, whereas the extended path method is best for utility functions II and IV. In the case of the German parameter set the parameterized expectations approach provides the best results for utility functions I through III and the extended path method for utility function IV.

Figure 1

Policy Functions

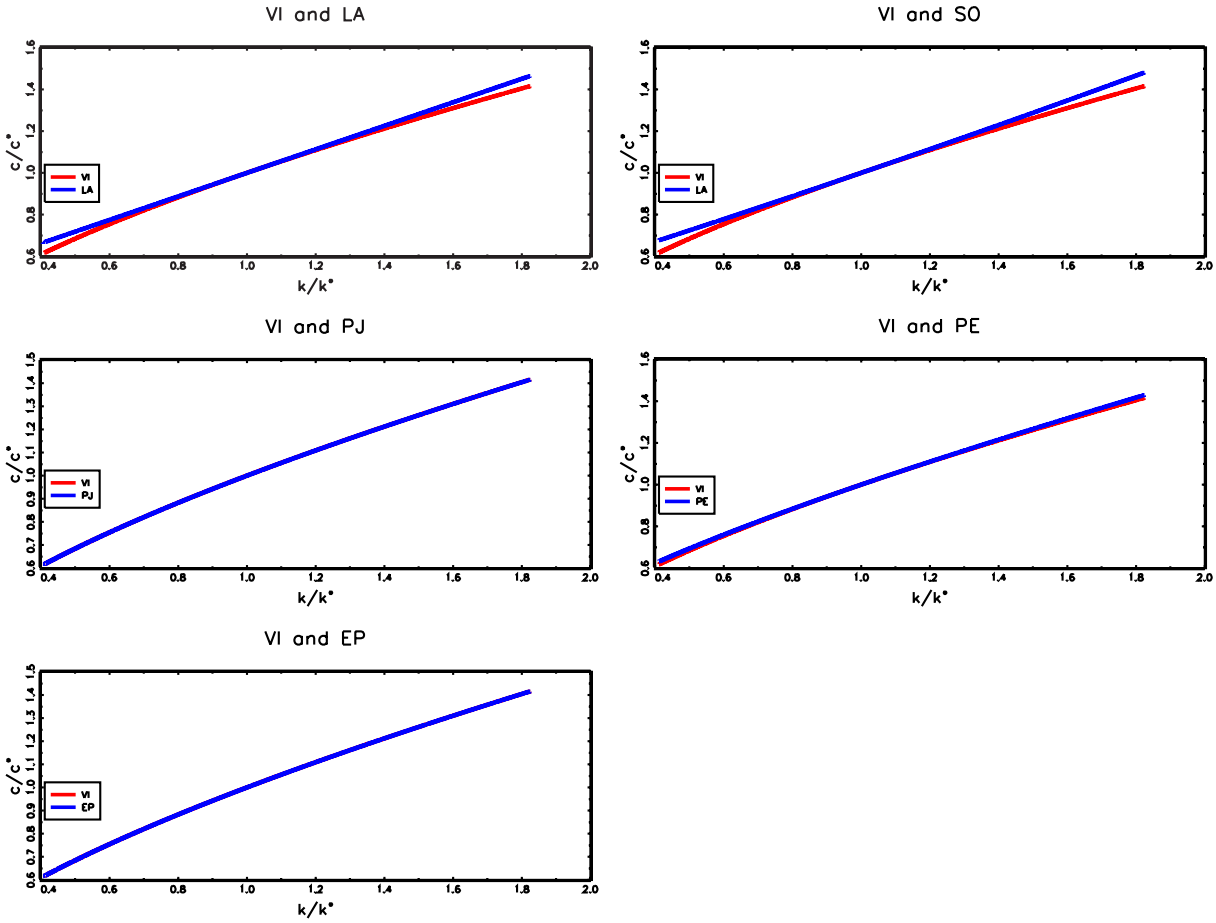

Figure 1 provides a second view on the different degrees of accuracy. It plots the policy function for consumption in the case of utility function I and the US parameter

set. Both the capital stock and consumption are measured relative to their respective stationary values. The upper left and right panel clearly document the poor performance of the log-linear and second-order approximation. The consumption function is only close to the solution obtained from value function iteration in the proximity of the stationary solution. The projection method as well as the extended path method, however, provide solutions that are visually indiscernible from the VI consumption function. Since the parameterized expectations approach relies on Monte Carlo integration it provides a good approximation in the range $[0.8k^*, 1.2k^*]$ but less so for values far apart from k^*. Yet, since virtually no simulated time series visits points in these regions, the PE provides a high degree of accuracy in terms of statistics computed from simulated time series.

Table 9

Summary Statistics: Utility I, US-Calibration, $\sigma = 0.0712$

Variable	s_x			s_{xy}			r_x		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Output	12.66	13.54	14.02	1.00	1.00	1.00	0.66	0.66	0.67
Standard Dev.	2.30	3.88	4.42	0.00	0.00	0.00	0.09	0.10	0.11
Investment	36.73	39.57	41.07	0.99	0.99	0.99	0.64	0.65	0.66
Standard Dev.	6.55	11.65	13.34	0.00	0.00	0.00	0.10	0.11	0.11
Consumption	4.30	4.48	4.60	0.90	0.90	0.91	0.76	0.76	0.76
Standard Dev.	0.93	1.26	1.41	0.02	0.03	0.03	0.08	0.09	0.10
Hours	5.99	6.03	6.10	0.96	0.96	0.98	0.64	0.65	0.66
Standard Dev.	0.99	1.02	1.07	0.01	0.02	0.03	0.09	0.10	0.11
Real Wage	6.89	7.29	7.52	0.98	0.98	0.98	0.69	0.69	0.70
Standard Dev.	1.32	2.04	2.32	0.00	0.01	0.01	0.09	0.10	0.11

Notes: See Table 3.

Sensitivity with respect to σ. The difference between the log-linear solution method and the other five methods considered in this paper is that the policy functions delivered by the former are independent of the size of the standard deviation of the productivity shock σ. To check the sensitivity of our results with respect to the size of σ we also consider the solutions of our model for utility function I and the US parameter set for $\sigma = 0.0712$, which amounts to ten times the size of the standard deviation of the log of the US-Solow residual commonly applied (see Table 1). In order to get results that are comparable to those in Tables 2 through 8 we do not change the parameters of the methods that determine their respective accuracy. For instance, we use the same

degree of the polynomials that approximate the rhs of the Euler equation for capital (9d) and the same number of grid-points in the value function iteration.

Table 9 reveals that our previous conclusion with respect to the sensitivity of the quantity allocations still holds: the difference between the largest and the smallest value of the standard deviations of output, investment, consumption, hours, and the real wage is always smaller than the average standard deviation of the respective statistic over 500 simulations of the model. Thus, even with an unrealistically volatile productivity shock, the researcher does not arrive at different conclusions about the nature of the business cycle if he/she arbitrarily uses any of the six methods.

Table 10

Selected Statistics $\sigma = 0.0712$, Utiltiy I, US-Calibration

Method	Run Time	Euler Equation Residual	Risk Free Rate of Return
LA	0.48 s	0.1841209	−2.310
SO	0.58 s	0.1787492	−3.175
PE	44 m 7.19 s	0.0053931	−2.753
GA	3 m 4.95 s	0.0055251	−2.691
EP	3 h 37m 44.81	0.0125591	−3.523
VI	5 h 33 m 19.09 s	0.0096766	−2.950

Notes:

See Table 2 for the abbreviation of methods, Table 8 for the definition of the Euler equation residual, and Table 7 for the definition and computation of the risk free rate of return. See also Table 2 for the definition of run time. The Euler equation residuals are the maximum absolute value from a grid of 2500 equally spaced points over the square $[0.6k^*, 2k^*] \times [0.7, 1.5]$. About 90 percent of the points of a long time series with 100,000 elements belong to this interval.

Table 10 shows that the ranking of the methods with respect to running time also remains unchanged. Value function iteration is still the most time-consuming method due to the large number of grid-points. Since we were unable to find a solution with the parameterized expectations method when we used the solution from the log-linear method to initialize the parameter vector, we had to use our genetic search algorithm instead. Therefore, the PE method is slow and the running time amounts to 44 minutes. The Galerkin projection algorithm, on the other hand, converges from the log-linear solution within about 3 minutes.

The great volatility of the productivity shock entails a large support of the stationary distribution of (k_t, Z_t). For this reason, we computed Euler equation residuals on 2,500 grid-points over the square $[0.6k^*, 2k^*] \times [0.7, 1.5]$. Since we have not taken any measures to increase the accuracy of our methods it comes as no surprise that all

Euler equation residuals are about an order of 10 higher than those reported in Table 8. Still, both local methods are the least accurate ones. However, the second-order approximation now performs a bit better vis-à-vis the log-linear solution in terms of the respective Euler equation residuals. The most accurate solutions are provided by the projection methods, where the parameterized expectations approach delivers an even better solution than the Galerkin projection method. Value function iteration and the extended path algorithm are in between the local methods and the projection methods, where the former has a slightly higher Euler equation residual than the latter.

The deviation of the mean risk-free rate of return from its stationary solution value is displayed in column four of Table 10. As in our benchmark calibration for the US economy there are no remarkable differences over the various methods. The largest deviation, again, occurs in the case of the extended path method, where we had to use a rather short time series of 10,000 points to save on computation time. In summary, our conclusions seem to be robust with respect to the size of the productivity shock.

5 Conclusion

This paper has shown that several numerical methods can be applied in order to study the standard business cycle model. Using either log-linearization, second-order approximation, parameterized expectations, Galerkin projection, deterministic extended path, or value function iteration basically results in the same values for the second moments of the variables that the business cycle researcher is most interested in, i.e. output, employment, investment, consumption, and wages. Log-linearization, of course, is very easy to implement and by far the fastest method. Furthermore, the solution from this method can often be successfully applied as an initial value for more sophisticated non-linear methods like parameterized expectations or Galerkin projection where the computation of a good initial value with genetic search algorithm or homotopy methods may become very time-consuming, as may be the case in more complex multi-dimensional state-space applications. Our results, therefore, suggest that the researcher may benefit from using log-linearization methods in the first place and, possibly, also should use non-linear methods such as parameterized expectations or projection methods to improve the accuracy of the computation in more non-linear problems that may arise, for example, in the presence of binding constraints, e.g. a non-negativity constraint on investment or a constraint on the maximum number of working hours, more curved utility functions, or models with sizable shocks.

Our work has also emphasized an important detail in the application of projection methods for the approximation of polynomial solution functions in more complex dynamic models. In our example of the stochastic growth model with flexible labor supply, standard projection methods failed to converge even for good initial values that were computed with the help of the log-linear solution. We find that the basic reason for this observation is the poor approximation of functions with Chebyshev polynomials outside the approximation interval. As a solution to this problem, we suggest to use a wider interval for the approximation of the function than for the integration over the residual function.

References

Altonij, J.G., 1986, Intertemporal Substitution in Labor Supply: Evidence from Micro Data, *Journal of Political Economics*, vol. 94, S17-S215.

Aruoba, S.B., J. Fernández-Villaverde, and J.F. Rubrio-Ramírez, 2006, Comparing Solution Methods for Dynamic Equilibrium Economies, *Journal of Economic Dynamics and Control*, forthcoming

Blanchard, O. J. and Ch. M. Kahn, 1980, The Solution of Linear Difference Models Under Rational Expectations, *Econometrica*, vol. 48, 1305-1311.

Castañeda, A., J. Díaz-Giminénez, J.-V. Ríos-Rull, 2004, Accounting for the US Earnings and Wealth Inequality, *Journal of Political Economy*, forthcoming.

Christiano, L.J., and J.D.M. Fisher, 2000, Algorithms for Solving Dynamic Models with Occasionally Binding Constraints, *Journal of Economic Dynamics and Control*, vol. 24, 1179-1232.

den Haan, W. J. and A. Marcet, 1994, Accuracy in Simulations, *Review of Economic Studies*, vol. 61, 3-17

Domeij, D., and M. Floden, 2006, The Labor Supply Elasticity and Borrowing Constraints: Why Estimates are Biased, *Review of Economic Dynamics*, forthcoming

Duffy, J., and P.D. McNelis, 2002, Approximating and Simulating the Stochastic Growth Model: Parameterized Expectations, Neural Networks, and the Genetic Algorithm, *Journal of Economic Dynamics and Control*, vol. 25, 1273-1303.

Greenwood, J., Z. Hercowitz, and G.W. Huffman, 1998, Investment, Capacity Utilization, and the Real Business Cycle, *American Economic Review*, vol. 78, 402-17.

Hansen, G., 1985, Indivisible Labor and the Business Cycle, *Journal of Monetary Economics*, vol. 16, 309-327.

Heer, B., and A. Maußner, 2005, *Dynamic General Equilibrium Modelling: Computational Methods and Applications*, Springer, Berlin

Heer, B., and M. Trede, 2003, 'Efficiency and Distribution Effects of a Revenue-neutral Income Tax Reform', *Journal of Macroeconomics*, vol. 25, 87-107.

Judd, K.L., 1992, Projection Methods for Aggregate Growth Models, *Journal of Economic Theory*, vol. 58, 410-52.

Judd, K.L., 1996, Approximation, Perturbation, and Projection Methods in Economic Analysis, in: H.M. Amman, D.A. Kendrick, and J. Rust, eds., *Handbook of Computational Economics*, vol. I, Elsevier Science B.V., Amsterdam.

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

Killingsworth, M.R., 1983, *Labor Supply*, Cambridge University Press, Cambridge, MA.

King, R. G., Ch. I. Plosser, and S. Rebelo, 1988, Production, Growth and Business Cycles I, The Basic Neoclassical Model, *Journal of Monetary Economics*, vol. 21, 195-232

King, R.G. and M.W. Watson, 2002, System Reduction and Solution Algorithms for Singular Linear Difference Systems under Rational Expectations, *Computational Economics*, vol. 20, 57-86

MaCurdy, T.E., 1981, An Empirical Model of Labor Supply in a Life-Cycle Setting, *Journal of Political Economy*, vol. 89, 1059-85.

McGrattan, E.R., 1999, Application of Weighted Residual Methods to Dynamic Economic Models, in: R. Marimon and A. Scott (Eds.), *Computational Methods for the Study of Dynamic Economies*, Oxford University Press: Oxford and New York, 114-142

Mitchell, M., 1996, *An Introduction to Genetic Algorithms*, Harvard University Press, Cambridge, MA.

Press, W.H., S. A. Teukolsky, W.T. Vetterling, and B.P. Flannery, 1992, Numerical Recipes in Fortran 77, Cambridge University Press: Cambridge

Schmitt-Grohé, S. and M. Uribe, 2004, Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function, *Journal of Economic Dynamics and Control*, vol. 28, 755-775

Sefton, J.A., 2000, A Solution Method for Consumption Decisions in a Dynamic Stochastic General Equilibrium Model, *Journal of Economic Dynamics and Control*, vol. 24, 1097-1119

Tauchen, G., 1986, Finite State Markov-Chain Approximations to Univariate and Vector Autoregressions, *Economics Letters*, vol. 20, 177-181.

Taylor, J.B., and H. Uhlig, 1990, Solving Nonlinear Stochastic Growth Models: A Comparison of Alternative Solution Methods, *Journal of Business and Economic Statistics*, vol. 8, 1-17.

Uhlig, H., 1999, A Toolkit for Analysing Nonlinear Dynamic Stochastic Models Easily, in: Ramon Marimon and Andrew Scott (Eds.), Computational Methods for the Study of Dynamic Economies, Oxford University Press: Oxford

Wright, B.D., and J.C. Williams, 1982, The Economic Role of Commodity Storage, *Economic Journal*, vol. 92, 596-614.

Wright, B.D., and J.C. Williams, 1984, The Welfare Effects of the Introduction of Storage, *Quarterly Journal of Economics*, vol. 99, 169-82.

Wright, B.D., and J.C. Williams, 1991, *Storage and Commodity Markets*, Cambridge University Press, Cambridge, UK.

Appendix

Value Function Iteration. Researchers that use value function iteration on a two-dimensional grid over the stock of capital and the realizations of the productivity shock face a trade-off between accuracy of the solution and computation speed. One way to alleviate this trade-off is interpolation between grid points (see, e.g., Sefton, 2000), another remedy is Howard's improvement algorithm or policy function iteration. This section motivates our procedure and provides the reasons why we think that it is superior to both interpolation and policy function iteration.

Consider the two-dimensional grid $\mathscr{G} := \mathscr{K} \times \mathscr{Z}$, over the stock of capital k, $\mathscr{K} = \{k_1, k_2, \ldots, k_n\}$, and the productivity shock z, $\mathscr{Z} = \{z_1, z_2, \ldots, z_m\}$. Let $P = (p_{jl})$ denote the transition matrix between states z_j and z_l that is associated with the Markov chain approximation of the productivity shock on $\mathscr{Z}$ as suggested by Tauchen (1986). The $n \times m$-matrix $V^s = (v_{ij}^s)$ stores the value function at iteration step s. Given the current state $(k_i, z_j) \in \mathscr{G}$, this allows us to write[17]

$$v_{ij}^{s+1} = \max_{k_h \in \mathscr{K},\, N} \quad \phi(k_h) := u(z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - k_h, N) + \beta a^{-\eta} \sum_{l=1}^{m} p_{jl} v_{hl}^s. \qquad (19)$$

In order to find the index of the maximal element k_h, we do not need to evaluate the expression on the rhs of this equation for all n elements of $\mathscr{K}$.[18] Since the policy function is monotonically increasing in k_i, we can start with the index of k found optimal for k_{i-1}. Furthermore, since $\phi(k_h)$ is strictly concave we are able to use a binary search algorithm that locates the maximal element after at most $\log_2 n$ evaluations of $\phi(k_h)$ (see Kremer, 2001, p. 166 for a proof). This is a considerable reduction of function evaluations. For instance, for $n = 1000$ elements we are done after 13 calls to the routine that returns $\phi(k_h)$ at the most.

If we do not want to use the Markov chain approximation of the productivity shock, equation (19) becomes

$$\begin{aligned} v_{ij}^{s+1} = \max_{k_h \in \mathscr{K},\, N} & u(z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - k_h, N) \\ & + \beta a^{-\eta} \int_{-\infty}^{\infty} v(k_h, \rho z_j + \epsilon)(2\pi\sigma^2)^{-1/2} e^{-(1/2\sigma^2)\epsilon^2} d\epsilon. \end{aligned} \qquad (20)$$

Gauss-Hermite quadrature is an efficient and accurate method to approximate the integral on the rhs of this equation. In our program we use four nodes. For this

[17] Note, that N is a function of k_h via (18).

[18] It seems that the literature has not paid ample attention to this point. See, for example the description of value function iteration on p. 41 of Ljungqvist and Sargent (2000).

method to be applicable we must interpolate the value function between the points of $\mathscr{Z}$. We employ a cubic spline. For this reason, each time we call the routine that evaluates the right-hand side of the max-operator we have to solve an m-dimensional linear tri-diagonal system of equations (see, e.g., Press, Teulkolsky, Vetterling, and Flannery, 1997, pp. 107-110). Not only does the interpolation step increase the computational time considerably, but also does the Gauss-Hermite integration decrease accuracy. Since the interval $[z_1, z_m]$ must contain the nodes of the Gauss-Hermite quadrature, we have to pick a rather large interval. With four nodes its size is roughly 21 times the unconditional standard deviation of the AR(1)-process (6). Even for simulations with as many as 100,000 realizations the productivity shock stays in a much smaller interval of about 9 times the size of the unconditional standard deviation of (6). Since the bounds of the grid for k depend on z_1 and z_m, we must also increase the interval $[k_1, k_n]$ to ensure that the policy function does not hit the boundary of $\mathscr{K}$. Given the same number of grid points as used in our preferred method the larger bounds introduce greater imprecision and the interpolation slows down the program. For instance, with $n = 10,000$, $m = 31$, and $z_m - z_1 = 9\sigma/\sqrt{1-\varrho^2})$ our method needs about 4 hours and 18 minutes to compute the policy function on a 3.2 GH Pentium 4 desk top computer. The accuracy of the solution as given by the maximum absolute value of the Euler equation residual computed over 400 equally spaced points in $[0.8k^*, 1.2k^*] \times [0.95, 1.05]$ amounts to 0.0002418.[19] When we use the interpolation method with Gauss-Hermite quadrature, me must decrease the number of grid points from $m = 31$ to $m = 15$ so that the program requires about the same time to compute the policy function (4 hours and 55 minutes). The respective Euler equation residual is 0.0009473. Thus, our method is faster and much more accurate!

A second approach to increase computation speed is Howard's improvement algorithm (see, e.g., Ljungqvist and Sargent, 2000, p. 32). Let $G = (g_{ij})$ denote a $n \times m$-matrix of integers that represent a feasible policy, i.e., g_{ij} is the index of $k_{g_{ij}} \in \mathscr{K}$ that is chosen if the current state of the system is $(k_i, z_j) \in \mathscr{G}$. The one-period utility associated with the policy matrix G is stored in the $n \times m$ matrix $U = (u_{ij})$:

$$u_{ij} = u(z_j N^{\alpha} k_i^{1-\alpha} + (1-\delta)k_i - k_{g_{ij}}, N).$$

If the policy G is maintained forever, the expected value of this policy at the grid $\mathscr{G}$ is

[19] k^* is the stationary stock of capital obtained from the deterministic version of the model. We compute the Euler equation residual using Gauss-Hermite quadrature with four nodes and interpolate bilinearly between the points of $\mathscr{G}$. Following Christiano and Fisher (2000), we compute the Euler residual as $\tilde{c}/c - 1$, where c is optimal consumption given the policy function. $\tilde{c}$ is the amount of consumption that is necessary to equate the lhs of the Euler equation for capital to its rhs.

the solution of the following system of linear equations:

$$v_{ij} = u_{ij} + \beta a^{-\eta} \sum_{l=1}^{m} p_{il} v_{g_{ij},l}, \quad i = 1, 2, \ldots, n;\ j = 1, 2, \ldots m. \tag{21}$$

Using the $n \times n$-matrices $H^j = (h^j_{il})$, $j = 1, 2, \ldots, m$, with

$$h^j_{il} = \begin{cases} 1 & \text{if } g_{ij} = l \\ 0 & \text{else,} \end{cases}$$

this system may be written as

$$\underbrace{\left[I_{nm \times nm} - \beta \begin{bmatrix} p_{11}H^1 & p_{12}H^1 & \ldots & p_{1m}H^1 \\ p_{21}H^2 & p_{22}H^2 & \ldots & p_{2m}H^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1}H^m & p_{2m}H^m & \ldots & p_{mm}H^m \end{bmatrix} \right]}_{=:A} \operatorname{vec} V = \operatorname{vec} U. \tag{22}$$

Howard's improvement algorithm starts with a feasible policy G, (1) solves (22) for V and (2) obtains a new policy by solving the maximization problem on the rhs of (19) with V given from the previous step. The algorithm repeats steps (1) and (2) until the policy function has converged. Different from value function iteration, which converges linearly, Howard's algorithm converges quadratically. However, the increased rate of convergence comes at the cost of solving the system (22). Note, that even for a moderately sized grid, the sparse matrix A is usually too large to fit into the memory of a conventional desk top computer. For instance, with $n = 5000$ and $m = 9$, A has 2.025 billion elements. Stored as double precision this matrix occupies more than 16 gigabytes of memory. Therefore, one must resort to sparse matrix methods in order to be able so solve this system. There is another disadvantage of this method which is specific to our problem. Since the AR(1)-process (6) is highly autocorrelated, many elements of the transition matrix P are close to zero so that the solution of (22) may be inaccurate. For a small grid, $n = 1000$ and $m = 9$, Howard's algorithm is about twice as fast as our method and about as precise. The former requires 96 seconds to converge and the associated maximum absolute value of the Euler equation residual equals 0.0023446. The latter needs 229 seconds to find the policy function and provides a maximum absolute value of the Euler equation residual of 0.0026142. For $n = 5000$ and $m = 9$ the increased time to solve the linear system (22) outweighs the higher rate of convergence. Our method finds the policy function in 27 minutes and 23 seconds whereas Howard's algorithm requires 33 minutes and 12 seconds to arrive at the same degree of accuracy of the solution. However, this result is sensitive with respect to the

size of the interval for the stock of capital. For the same set of parameters but a smaller interval $[k_1, k_n]$ our method is more than 37 times faster than policy function iteration, which needs more than one day and two hours to find a solution of equal accuracy as our algorithm. Thus, Howard's improvement algorithm is a good choice if the shock is a Markov chain with few elements and a transition matrix with elements well above zero, but not the method of choice for the stochastic growth model.