Solving Asset Pricing Models with Stochastic Dynamic Programming *

Lars Grüne
Mathematisches Institut
Fakultät für Mathematik und Physik
Universität Bayreuth
95440 Bayreuth, Germany
lars.gruene@uni-bayreuth.de

Willi Semmler
Center for Empirical Macroeconomics
Bielefeld University and
New School University, New York
wsemmler@wiwi.uni-bielefeld.de

December 15, 2003

Abstract: The study of asset price characteristics of stochastic growth models such as the risk-free interest rate, equity premium and the Sharpe ratio has been limited by the lack of global and accurate methods to solve dynamic optimization models. In this paper a stochastic version of a dynamic programming method with adaptive grid scheme is applied to compute the above mentioned asset price characteristics of a stochastic growth model. The stochastic growth model is of the type as developed by Brock and Mirman (1972) and Brock (1979, 1982). In order to test our method it is applied to a basic stochastic growth model for which the optimal consumption and asset prices can analytical be computed. Since, as shown, our method produces only negligible errors as compared to the analytical solution it is recommended to be used for more elaborate stochastic growth models with different preferences and technology shocks, adjustment costs, and heterogenous agents.

JEL Classification: C60, C61, C63, D90, G12

Keywords: stochastic growth models, stochastic dynamic programming, adaptive grid, asset pricing

*We want to thank Buz Brock, John Cochrane, Martin Lettau, Manuel Santos and Ken Judd for helpful communications. We are also grateful for comments from the participants of the SCE conference, Seattle, July 2003.

1 Introduction

Asset pricing in intertemporal models with exogenous dividend stream[1] have had difficulties to match financial market characteristics such the risk-free interest rate, equity premium and the Sharpe-ratio, a measure of the risk-return trade-off. In those models the risk-free rate turns out to be too high and the mean equity premium and Sharpe-ratio too low compared to what one finds in time series data.[2]

There is another tradition of asset pricing models that is based on the stochastic growth model with production originating in Brock and Mirman (1972) and Brock (1979, 1982). The Brock approach extends the asset pricing strategy beyond endowment economies to economies that have endogenous state variables including capital stocks that are used in production. Authors, building on this tradition,[3] have argued that it is crucial how consumption is modelled. In stochastic growth models the randomness occurs to the production function of firms and consumption and dividends are derived endogenously. Yet, the latter type of models have turned out to be even less successful. Given a production shock, consumption can be smoothed through savings and thus asset market facts are even harder to match.[4]

After those findings two basic issues arose. The first issue concerns the accuracy of the solutions of the stochastic growth model. One conjecture in the literature was thus that the solution of stochastic growth models through linearizations for the cases where the value function may have steep curvature or possibly is non-smooth may not be appropriate. A detailed discussion of those issues is taken up in sect. 2 of the paper. Recently, global solution techniques such as global solutions to the Hamilton-Jacobi-Bellman equation have been developed that can address this concern. The latter global solution method uses stochastic dynamic programming with discretization of the state space and adaptive gridding strategy to obtain more accurate solutions.[5] Again, a full discussion of the literature is given in sect. 2.

Another major issue concerning the problem to what extend the stochastic growth model can match real and financial time series characteristics is related to the simple structure of the basic model. In order to allow to match asset price characteristics with data economic research has extended the standard stochastic growth model to include different utility functions, in particular habit formation, adjustment cost of capital, a two sector economy, or heterogenous firms and households (see Cochrane 2001, ch. 21)).[6]

[1] Those models originate in Lucas (1978) and Breeden (1979) for example.

[2] This results from the too low variability of consumption as compared to the high variability of asset returns in the time series data.

[3] See Rouwenhorst (1995, Akdeniz and Dechert (1997), Jerman (1998), Boldrin, Christiano and Fisher (2001), Lettau and Uhlig (1999) and Hansen and Sargent (2002), the latter in a linear-quadratic economy. The Brock model has also been used to evaluate the effect of corporate income tax on asset prices, see McGrattan and Prescott (2001).

[4] For a recent account of the gap between models and facts, see Boldrin, Christiano and Fisher (2001), Cochrane (2001, ch. 21), Lettau, Gong and Semmler (2001) and Semmler (2003, chs. 9-10).

[5] For deterministic versions, see Grüne (1997), Santos and Vigo–Aguiar (1998), and Grüne and Semmler (2003).

[6] For further detailed studies see, for example, Campbell and Cochrane (1999), Jerman (1998), Boldrin, Christiano and Fisher (2001) for extensions along those lines.

In this paper, we are mainly concerned with the first set of issues. Yet, once we can have sufficient confidence on the accuracy of the stochastic dynamic programming method, it is easily applicable to extended models. In our method we do not use fixed grids, but adaptive space discretization. In some of the literature, see Munos and Moore (2002) and Trick and Ziu (1997), adaptive methods for dynamic programming have been used, but those methods either do not allow the derivation of rigorous error bounds or are computationally very expensive, cf. the discussion in Section 2. In the method applied here efficient and reliable local error estimation is undertaken and used as a basis for a local refinement of the grid in order to deal with regions steep slopes of the value function. This procedure allows for a global dynamic analysis of deterministic as well as stochastic intertemporal decision problems.

In order to test the stochastic dynamic programming algorithm with flexible grid size we numerically provide the global solution and dynamics of the most basic stochastic growth model, as based on Brock and Mirman (1972) and Brock (1978, 1982). The model can analytical be solved for the policy function in feedback form. Moreover, the asset prices, the risk-free interest rate, the equity premium and the Sharpe-ratio, can, once the model is solved analytically for the sequence of optimal consumption, easily be solved numerically and those solutions compared to the numerical solutions obtained by our stochastic dynamic programming method. As will be shown the errors, as compared to the analytical solutions, are negligibly small. Thus, the method we propose here and test for the basic stochastic growth model can easily be applied to the extensions of the basic model recently proposed in the literature.

The paper is organized as follows. Section 2 discusses related literature. Section 3 presents the stochastic dynamic programming algorithm. Section 4 introduces the stochastic discount problem in asset pricing and the measures of the financial characteristics we want to study and provides the numerical steps by which we obtain the financial characteristics. Section 5 applies those steps to the numerically specified basic growth model. Section 6 concludes the paper.

2 Related Literature

In the literature one can find a vast amount of different approaches to dynamic programming problems, many of them using state–of–the art mathematical and numerical techniques for making this appealing global approach more efficient. In this paper we apply an adaptive gridding algorithm, see Section 3 for details. In the present section we briefly review some other approaches and highlight differences to and similarities with our approach.

One of the fundamental difficulties with the dynamic programming approach is that the computational load grows exponentially with the dimension of the problem, a phenomenon known as the "curse of dimensionality" (see Rust (1996) for a comprehensive account on complexity issues). In our case, for computing asset pricing in the context of stochastic growth models, starting with Brock and Mirman (1972) as suggested in the literature, the problem to be solved is two dimensional, hence this is not a crucial aspect here. Nevertheless, for the sake of completeness we want to mention approaches like randomly distributed grid points (Rust (1997)) or so called low discrepancy grids (Rust (1996), Reiter (1999)) which are able to break the curse of dimensionality. In principle also Monte–Carlo techniques like in Keane and Wolpin (1994) allow for breaking the curse of dimensionality, but as Rust (1997) points out, the specific algorithm in Keane and Wolpin (1994) uses an interpolation technique which again is subject to exponential growth of the numerical cost in the space dimension.

For low dimensional problems the goal of the numerical strategy is not to avoid the curse of dimensionality but rather to reduce the computational cost for a problem of fixed dimension. For this purpose, two main approaches can be found in the literature, namely higher order approximations and adaptive gridding techniques; the latter will be used in our numerical approach.

The idea of high order approximations lies in exploiting the smoothness of the optimal value function: if the optimal value function turns out to be sufficiently smooth, then methods using approximations by smooth functions, like Chebyshev polynomials (Rust (1996), Judd (1996), Jermann (1998)), Splines (Daniel (1976), Johnson et al. (1993), Trick and Zin (1993, 1997)) or piecewise high–order approximations (Falcone and Ferretti (1998)) can be very efficient. Smoothness is also the basis of other high–order strategies, like in finite difference approximations (Candler (2001)), Gaussian Quadrature discretization (Tauchen and Hussey (1991), Burnside (2001)) and in perturbation techniques (Judd (1996)). Yet, the last should also work if the value function is only piecewise smooth.[7]

Some of these methods (like Spline and piecewise high order approximation) use a (fixed) grid discretization of the state space similar to our approach. The combination of adaptive grids with higher order approximation is currently under investigation and it will be interesting to see whether adaptive discretization ideas based on our local error estimation technique work equally well with these approximation techniques.

Concerning discretization techniques it should be noted that from the complexity point of view it turns out to be optimal to solve the dynamic programming problem on successively finer grids, using a one–way multigrid strategy (Chow and Tsitsiklis (1991), see also Rust (1996)). In fact, our adaptive gridding algorithm is similar to this approach since the approximation on the previous grid Γ_i is always used as the initial value for the computation on the next finer adaptive grid Γ_{i+1}. This also explains the large reduction in computation time observed for our approach compared to the computation on *one* fixed equidistant grid.

Let us now turn to the methodology employed here, i.e., adaptive gridding techniques. Perhaps closest to our approach are the techniques discussed in Munos and Moore (2002). Here a number of heuristic techniques are compared which lead to local and global error indicators which can in turn be used for an adaptive grid generation. Some of the indicators discussed in this paper bear some similarity with our residual based estimator, though rigorous estimates as given in sect. 3.3. of our paper are not given there. In any case, the authors report that these techniques are unsatisfactory and argue for a completely different approach which measures the influence of local errors in certain regions on the global error by analyzing the information flow on the Markov chain related to the

[7]For an early survey of those methods, see Taylor and Uhlig (1990) where one can find a comparative numerical study of several methods.

discretization of the (deterministic) problem at hand. The reason for this lies in the fact that the model problem treated by Munos and Moore (2002) has a discontinuous optimal value function, which often happens in technical problems with boundary conditions. In fact, also our adaptive scheme performs rather poorly in presence of discontinuities but since our economic problems do always have continuous optimal value functions, Munos' and Moore's conclusions do not apply here. A roughly similar technique is the endogenous oversampling used by Marcet (1994). This is again a heuristic method, which, however, does not lead to adaptive grids but rather selects suitable parts of the state space where the optimally controlled trajectories stay with high probability.

Probably the adaptive approaches with the most solid mathematical background are presented in the papers of Trick and Zin (1993, 1997).[8] In these papers an alternative approach for the solution of the fully discrete problem is developed using advanced linear programming techniques which are capable of solving huge linear programs with many unknowns and constraints. In Trick and Zin (1993) an adaptive selection of constraints in the linear program is used based on estimating the impact of the missing constraint, a method which is closely related to the chosen solution method but only loosely connected to our adaptive gridding approach. The later paper (Trick and Zin (1997)), however, presents an idea which is very similar to our approach. Due to the structure of their solution they can ensure that the numerical approximation is greater than or equal to the true optimal value function. On the other hand, the induced suboptimal optimal control strategy always produces a value which is lower than the optimal value. Thus, comparing these values for each test point in space one can compute an interval in which the true value must lie, which produces a mathematically concise error estimate that can be used as a refinement criterion. While this approach is certainly a good way to measure errors, which could in particular be less conservative than our measure for an upper bound, we strongly believe that it is less efficient for an adaptive gridding scheme, because (i) the estimated error measured by this procedure is not a local quantity (since it depends on the numerical along the whole suboptimal trajectory), which means that regions may be refined although the real error is large elsewhere, and (ii) compared to our approach it is expensive to evaluate, because for any test point one has to compute the whole suboptimal trajectory, while our residual based error estimate needs only one step of this trajectory.

Let us comment on the idea of a posteriori error estimation. In fact, the idea to evaluate residuals can also be found in the papers of Judd (1996) and Judd and Guu (1997), using, however, not the dynamic programming operator but the associated Euler equation. In these references the resulting residual was used to estimate the quality of the approximating solution, but to our knowledge it has not been used to control adaptive gridding strategies, and we are not aware of any estimates such as ours which is a crucial property for an efficient and reliable adaptive gridding scheme.

Summarizing our discussion, there are a number of adaptive strategies around which are all reported to show good results, however, they are either heuristic[9] and better suited

[8]As mentioned above, this approach also uses splines, i.e., a smooth approximation, but the ideas developed in these papers do also work for linear splines which do not require smoothness of the approximated optimal value function.

[9]In order to avoid misunderstandings: We do not claim that heuristic methods cannot perform well; in fact they can show very good results. Our main concern about these methods is that one can never be sure about the quality of the final solution of a heuristic method.

for other classes of problems or they have nice theoretical features but are practically inconvenient because their implementation is numerically much more expensive than our approach.

3 Stochastic Dynamic Programming

Next we describe the stochastic numerical dynamic programming algorithm that we use to solve the asset pricing characteristics of the stochastic growth model.

3.1 Preliminaries

We consider the discrete stochastic dynamic programming equation

$$V(x) = \max_{c \in C} E\{u(x,c,\varepsilon) + \beta(x,\varepsilon)V(\varphi(x,c,\varepsilon))\}. \tag{3.1}$$

Here $x \in \Omega \subset \mathbb{R}^2$, $C \subset \mathbb{R}$, Ω and C are compact sets and ε is a random variable with values in $\mathbb{R}$. The mappings $\varphi : \Omega \times C \times \mathbb{R} \to \mathbb{R}^2$ and $g : \Omega \times C \times \mathbb{R} \to \mathbb{R}$ are supposed to be continuous and Lipschitz continuous in x. Furthermore, we assume that either $\varphi(x,c,z) \in \Omega$ almost surely for all $x \in \Omega$ and all $c \in C$, or that suitable boundary values $V(x)$ for $x \notin \Omega$ are specified, such that the right hand side of (3.1) is well defined for all $x \in \Omega$. The value $\beta(x,\varepsilon)$ is the (possibly state and ε dependent) discount factor which we assume to be Lipschitz and we assume that there exists $\beta_0 \in (0,1)$ such that $\beta(x,\varepsilon) \in (0,\beta_0)$ holds for all $x \in \Omega$. We can relax this condition if no maximization takes place, in this case it suffices that all trajectories end up in a region where $\beta(x,\varepsilon) \in (0,\beta_0)$ holds. This is the situation for the asset price problem, cf. the discussion in Cochrane (2001), p. 27.

Associated to (3.1) we define the dynamic programming operator

$$T : C(\Omega,\mathbb{R}) \to C(\Omega,\mathbb{R})$$

given by

$$T(W)(x) := \max_{c \in C} E\{u(x,c,\varepsilon) + \beta(x,\varepsilon)W(\varphi(x,c,\varepsilon))\}. \tag{3.2}$$

The solution V of (3.1) is then the unique fixed point of (3.2), i.e.,

$$T(V) = V. \tag{3.3}$$

For the numerical solution of (3.3) we use a discretization method that goes back to Falcone (1987) in the deterministic case and was applied to stochastic problems in Santos and Vigo–Aguiar (1998). Here we use unstructured rectangular grids: We assume that $\Omega \subset \mathbb{R}^n$ is a rectangular and consider a grid Γ covering Ω with rectangular elements Q_l and nodes x_j and the space of continuous and piecewise multilinear functions

$$\mathcal{W}_\Gamma := \{W \in C(\Omega,\mathbb{R}) \,|\, W(x + \alpha e_j) \text{ is linear in } \alpha \text{ on each } Q_l \text{ for each } j = 1,\ldots,n\}$$

where the e_j, $j = 1, \dots, n$ denote the standard basis vectors of the $\mathbb{R}^n$, see Grüne (2003) for details of the grid construction. With $\pi_\Gamma : C(\Omega, \mathbb{R}) \to \mathcal{W}_\Gamma$ we denote the projection of an arbitrary continuous function to $\mathcal{W}_\Gamma$, i.e.,

$$\pi_\Gamma(W)(x_j) = W(x_j) \text{ for all nodes } x_j \text{ of the grid } \Gamma.$$

Note that our approach easily carries over to higher order approximations, the use of multilinear approximations is mainly motivated by its ease of implementation, especially for adaptively refined grids.[10] Also, the approach can easily be extended to higher dimensions.

We now define the discrete dynamic programming operator by

$$T_\Gamma : C(\Omega, R) \to \mathcal{W}_\Gamma, \quad T_\Gamma = \pi_\Gamma \circ T \tag{3.4}$$

with T from (3.2). Then the discrete fixed point equation

$$T_\Gamma(V_\Gamma) = V_\Gamma. \tag{3.5}$$

has a unique solution $V_\Gamma \in \mathcal{W}_\Gamma$ which converges to V if the size of the elements Q_l tends to zero. The convergence is linear if V is Lipschitz on Ω, see Falcone (1987), and quadratic if V is C^2, see Santos and Vigo–Aguiar (1998).

3.2 Numerical evaluation of T_Γ and solution of (3.5)

For the solution of (3.5) as well as for the computation of $\eta(x)$ we need to evaluate the operator T_Γ. More precisely, we need to evaluate

$$\max_{c \in C} E\{u(x_j, c, \varepsilon) + \beta(x_j, \varepsilon) W(\varphi(x_j, c, \varepsilon))\}.$$

for all nodes x_j of Γ.

This first includes the numerical evaluation of the expectation E. If ε is a finite random variable then this is straightforward, if ε is a continuous random variable then the corresponding integral

$$\int (u(x, c, \varepsilon) + \beta(x, \varepsilon) V(\varphi(x, c, \varepsilon))) f(\varepsilon) d\varepsilon$$

has to be computed, where f is the probability density of ε. In our implementation we approximated this integral by a trapezoidal rule with 10 equidistant intervals.

The second difficulty in the numerical evaluation of T lies in the maximization over c. In our implementation we simply used a discrete approximation of the set C with equidistant points and then maximized by comparing the corresponding values. For low to medium accurate evaluation of T_Γ this is a feasible method, for high accuracy, however, this method becomes inefficient; in this case other methods like, e.g., Brent's algorithm have been reported in the literature to give good results.

For the solution of the fixed point equation (3.5) we use the Gauss–Seidel type value space iteration which is described in Section 3 of Grüne (1997) (under the name "increasing

[10] The combination of adaptive grids and higher order approximations is currently under investigation.

coordinate algorithm"), where we subsequently compute $V_{i+1} = S_\Gamma(V_i)$ with S_Γ being a Gauss–Seidel type iteration operator (including the maximization over c) obtained from T_Γ. This iteration is coupled with a policy space iteration: Once a prescribed percentage of the maximizing u–values in the nodes remains constant from one iteration to another we fix all control values and compute the associated value function by solving a linear system of equations using the CGS or BICGSTAB method (in our examples the CGS method turned out to show more reliable convergence behavior). After convergence of this method we continue with the value space iteration using S_Γ until the control values again converge, switch to the linear solver and so on. This combined policy–value space iteration turns out to be much more efficient (often more than 90 percent faster) than the plain Gauss–Seidel value space iteration using S_Γ, which in turn is considerably faster than the Banach iteration $V_{i+1} = T_\Gamma(V_i)$.

3.3 Error estimation

The basic idea of our adaptive gridding algorithm lies in evaluating the residual of the operator T applied to V_Γ, as made precise in the following definition. Here for any subset $B \subset \Omega$ and any function $W \in C(\Omega, \mathbb{R})$ we use

$$\|W\|_{\infty,B} := \max_{x\in B} |W|.$$

(i) We define the *a posteriori error estimate* η as a continuous function $\eta \in C(\Omega, \mathbb{R})$ by

$$\eta(x) := |T(V_\Gamma)(x) - V_\Gamma(x)|.$$

(ii) For any element Q_l of the grid Γ we define the *elementwise error estimate*

$$\eta_l := \|\eta\|_{\infty,Q_l}$$

(iii) We define the *global error estimate* $\eta_{\max}$ by

$$\eta_{\max} := \max_l \eta_l = \|\eta\|_\infty.$$

It is shown in Grüne (2003), that for this error estimate the inequalities

$$\frac{\eta_{\max}}{1+\beta_0} \le \|V - V_\Gamma\|_\infty \le \frac{\eta_{\max}}{1-\beta_0}$$

holds. These inequalities show that the error estimate is *reliable* and *efficient* in the sense of numerical error estimator theory, which is extensively used in the numerical solution of partial differential equations. Furthermore, $\eta(x)$ is continuous and one can show that a similar upper bound holds for the error in the derivative of V and V_Γ.

If the size of a grid element tends to zero then also the corresponding error estimate tends to zero, even quadratically in the element size if V_Γ satisfies a suitable "discrete C^2" condition, i.e., a boundedness condition on the second difference quotient.

This observation shows that refining elements carrying large error estimates is a strategy that will eventually reduce the element error and consequently the global error, and thus forms the basis of the adaptive grid generation method which we will describe in the next section.

Clearly, in general the values $\eta_l = \max_{x \in Q_l} \eta(x)$ can not be evaluated exactly since the maximization has to be performed over infinitely many points $x \in Q_l$. Instead, we approximate η_l by

$$\tilde{\eta}_l = \max_{x_T \in X_T(Q_l)} \eta(x_T),$$

where $X_T(Q_l)$ is a set of test points. In our numerical experiments we have used the test points indicated in Figure 3.1.

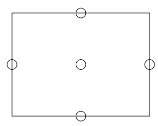

Figure 3.1: Test points $X_T(Q_l)$ for a 2d element Q_l

3.4 Adaptive refinement and coarsening

The adaptive grid itself was implemented on a tree data structure in the programming language C. The adaptive refinement follows the standard practice in numerical schemes and works as follows:

(0) Choose an initial grid Γ_0, set $i = 0$, fix a refinement threshold $\theta \in (0, 1)$

(1) Compute V_{Γ_i} and the (approximated) error estimates $\tilde{\eta}_l$ and $\tilde{\eta}_{\max}$. If a desired accuracy or a maximally allowed number of nodes is reached, then stop

(2) Refine all elements Q_l with $\tilde{\eta}_l \geq \theta \tilde{\eta}_{\max}$, denote the new grid by Γ_{i+1}

(3) Set $i := i + 1$ and go to (1)

Here for the solution of V_{Γ_i} for $i \geq 1$ we use the previous solution $V_{\Gamma_{i-1}}$ as the initial value for the iteration described in Section 3.2, which turns out to be very efficient.

During the adaptation routine it might happen that the error estimate causes refinements in regions which later turn out to be very regular. It is therefore advisable to include a coarsening mechanism in the above iteration. This mechanism can, e.g., be controlled by comparing the approximation V_{Γ_i} with its projection $\pi_{\widetilde{\Gamma}_i} V_{\Gamma_i}$ onto the grid $\widetilde{\Gamma}_i$ which is obtained from Γ_i by coarsening each element once. Using a specified coarsening tolerance $tol \geq 0$ one can add the following step after Step (2).

(2a) Coarsen all elements Q_l with $\tilde{\eta}_l < \theta \tilde{\eta}_{\max}$ and $\|V_{\Gamma_i} - \pi_{\widetilde{\Gamma}_i} V_{\Gamma_i}\|_{\infty, Q_l} \leq tol$.

This procedure also allows to start from rather fine initial grids Γ_0, which have the advantage of yielding a good approximation $\tilde{\eta}_l$ of η_l. Unnecessarily fine elements in the initial grids will this way be coarsened afterwards.

In addition, it might be desirable to add additional refinements in order to avoid large differences in size between adjacent elements, e.g., to avoid degeneracies. Such regularization steps could be included as a step (2b) after the error based refinement and coarsening has been performed. In our implementation such a criterion was used; there the difference in refinement levels between two adjacent elements was restricted to at most one. Note that the values in the hanging nodes (these are the nodes appearing at the interface between two elements of different refinement level) have to be determined by interpolation in order to ensure continuity of V_Γ.

In addition, our algorithm allows for the anisotropic refinement of elements: consider an element Q of Γ (we drop the indices for notational convenience) and let $X_{new,i}$ be the set of potential new nodes which would be added to Γ if the element Q_l was refined in coordinate direction e_i, cf. Figure 3.2.

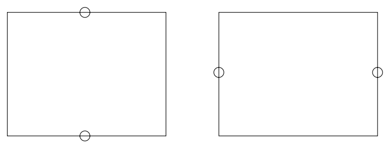

Figure 3.2: Potential new nodes $X_{new,1}$ (left) and $X_{new,2}$ (right) for a 2d element Q

Define the error estimate in these nodes for each coordinate direction e_i by $\eta_{dir,i} := \max_{x \in X_{new,i}} \eta(x)$ and define the overall error measured in these potential new nodes by $\eta_{dir} := \max_{i=1,\dots,n} \eta_{dir,i}$. Note that $\eta_{dir} \leq \eta_l$ always holds. If we include all the points in $X_{new} := \bigcup_{i=1,\dots,n} X_{new,i}$ in our set of test points $X_T(Q)$ (which is reasonable because in order to compute $\eta_{dir,i}$ we have to evaluate $\eta(x)$ for $x \in X_{new}$, anyway) then we can also ensure $\eta_{dir} \leq \tilde{\eta}_l$.

Now we refine the element only in those directions for which the corresponding test points yield large values, i.e., if the error estimate $\eta_{dir,1}$ is large we refine in x–direction and if the error estimate $\eta_{dir,2}$ is large we refine in y–directions (and, of course, we refine in both directions if all test points have large error estimates).

Anisotropic refinement can considerably increase the efficiency of the adaptive gridding strategy, in particular if the solution V has certain anisotropic properties, e.g., if V is linear or almost linear in one coordinate direction. Note that this is the case in our example and the anisotropic refinement is clearly visible in Figure 3.1. On the other hand, a very anisotropic grid Γ can cause degeneracy of the function V_Γ like, e.g., large Lipschitz constants or large (discrete) curvature even if V is regular, which might slow down the convergence. However, according to our numerical experience the positive effects of anisotropic grids are usually predominant.

4 The Stochastic Decision Problem in Asset Pricing

Our stochastic decision problem arising from the stochastic growth model in the Brock tradition which we want to solve and to which we want to compute certain financial measures is as follows. We want to solve the optimal control c_t for the optimal control problem

$$V(k,z) = \max_{c_t} E\left(\sum_{t=0}^{\infty} \beta^i u(c_t)\right) \tag{4.1}$$

subject to the dynamics

$$\begin{aligned} k_{t+1} &= \varphi_1(k_t, z_t, c_t, \varepsilon_t) \\ z_{t+1} &= \varphi_2(k_t, z_t, c_t, \varepsilon_t) \end{aligned}$$

using the constraints $c_t \geq 0$ and $k_t \geq 0$ and the initial value $k_0 = k, z_0 = z$. Here $(k_t, z_t) \in \mathbb{R}^2$ is the state and ε_t are i.i.d. random variables. We abbreviate $x_t = (k_t, z_t)$ and $\varphi(x, c, \varepsilon) = (\varphi_1(k, z, c, \varepsilon), \varphi_2(k, z, c, \varepsilon))$, i.e.,

$$x_{t+1} = \varphi(x_t, c_t, \varepsilon_t).$$

This optimal control problem allows the computation of c in feedback form, i.e. $c_t = c(x_t)$ for some suitable map $c : \mathbb{R}^2 \to \mathbb{R}$.

Based on this c we compute the stochastic discount factor[11]

$$m(x_t) = \beta \frac{u'(c(x_{t+1})}{u'(c(x_t))} \tag{4.2}$$

(note that m depends on ε_t), which serves as an ingredient for the next step, which consists of solving the asset pricing problem

$$p(x) = E\left(\sum_{t=1}^{\infty} \prod_{s=1}^{t} m(x_s) d(x_t)\right), \tag{4.3}$$

where $d(x_t)$ denotes the dividend at x_t and $x_0 = x$ and the dynamics are given by

$$x_{t+1} = \varphi(x_t, c(x_t), \varepsilon_t)$$

with c from above.

Finally, we use these values to compute the Sharpe ratio

$$S = \left| \frac{E(R(x)) - R^f(x)}{\sigma(R(x))} \right| = \frac{-R^f(x) \mathrm{cov}\Big(m(x), R(x)\Big)}{\sigma(R(x))} \tag{4.4}$$

and its upper bound

$$S_B = \frac{\sigma(m(x))}{E(m(x))}. \tag{4.5}$$

Here

$$R^f(x) = \frac{1}{E(m(x))} \tag{4.6}$$

[11]The following financial measures are introduced and studied in detail in Cochrane (2001).

is the risk free interest rate and

$$R(x_t) = \frac{d(x_{t+1}) + p(x_{t+1})}{p(x_t)} \tag{4.7}$$

is the gross return.

Note that the equality $E(m(x)R(x)) = 1$ holds, which can serve as a indicator for the accuracy of our numerical solution.

We solve the asset pricing problem in the following three steps:

(i) We compute the optimal value function V of the underlying optimal control problem, and compute c from V

(ii) We compute the prices $p(x)$ from c and m

(iii) We compute the Sharpe ratio S and its bound S_B from c, m and p

For our test model, both c and p are actually available analytically. This allows us to test the single steps of our algorithm by replacing the numerically computed c in (ii) and (iii) and/or p in (iii) by their exact values.

For each of the steps we do now sketch our technique for the numerical computation using the algorithm described above in Section 3.

Step (i):

For the solution of the optimal control problem we use a dynamic programming algorithm with adaptive grid. In order to solve (4.1) we solve the equivalent dynamic programming equation

$$V(x) = \max_c E\left(u(c) + \beta V(\varphi(x, c, \varepsilon))\right) =: T(V)(x) \tag{4.8}$$

For solving this equation we choose a computational domain $\Omega \subset \mathbb{R}^2$ and approximate V on a rectangular grid Γ covering Ω, using multilinear interpolation between the grid nodes. For continuous and piecewise multilinear function V_Γ on Γ, we solve the fixed point equation

$$V_\Gamma(x_i) = T(V_\Gamma)(x_i) \tag{4.9}$$

for all nodes x_i of Γ.

In order to solve (4.9) we use a mixed value/policy iteration method: We first iterate (4.9) (using a Gauss–Seidel like acceleration method) until we observe convergence of the maximizing control values c. Then we fix the values and solve the resulting system of linear equations (using the CGS method). Iteratively, we continue this process until convergence.

Here in each step of the iteration we have to perform a maximization over c and an evaluation of the expectation. Knowing the density function $p(\varepsilon)$ of the underlying random variable ε, the latter problem lies in evaluating the integral

$$\int V(\varphi(t, x, c, \varepsilon))p(\varepsilon)d\varepsilon,$$

which can be efficiently accomplished by a numerical quadrature rule.

The maximization turns out to be a more severe numerical problem. In our implementation we have used a simple and straightforward method by discretizing the set C of possible values of c and maximizing by comparing the finitely many discrete values.[12]

Another crucial point in solving (4.9) is the choice of an appropriate grid. Here we make use of an adaptive gridding strategy. After the solution V_Γ is computed, we evaluate the error estimates

$$\eta(x) = |V_\Gamma(x) - T(V_\Gamma)(x)|.$$

This value gives an upper and lower bound for the real global error. Thus, we evaluate η in a number of test points in each grid element and refine those elements carrying a large error estimate. This way, we can iteratively construct a grid which is adjusted to the problem.

Once V_Γ is computed with sufficient accuracy we can obtain the optimal control value $c(x)$ in each point by choosing $c(x)$ such that (4.8) is maximized, i.e., such that

$$E\left(u(c(x)) + \beta V_\Gamma(\varphi(x, c(x), \varepsilon))\right) = \max_c E\left(u(c) + \beta V_\Gamma(\varphi(x, c, \varepsilon))\right)$$

holds. Once c is known, m of equ. (4.2) can be computed from this value.

Step (ii):

For computing $p(x)$ we follow the same approach as in Step (i), except that here $c(x)$ is known in advance and hence no maximization needs to be done.

For the computation of p we first solve the dynamic programming equation

$$\tilde{p}(x) = E(d(x) + m(x)p(\varphi(x, c(x), \varepsilon)))$$

which is simply a system of linear equations which we solve using the CGS method. This yields a numerical approximation of the function

$$\tilde{p}(x) = E\left(\sum_{t=0}^{\infty} \prod_{s=1}^{t} m(x_s) d(x_t)\right)$$

(with the convention $\prod_{s=1}^{0} m(x_s) = 1$), from which we obtain p by

$$p(x) = \tilde{p}(x) - d(x).$$

In our numerical computations for the computation of p we have always used the same grid Γ as in the previous computation of V in Step (i). The reason is that it did not seem justified to use a finer grid here, because the accuracy of the entering values c from Step (i) is limited by the resolution of Γ, anyway. However, it might nevertheless be that using a different grid (generated e.g. by an additional adaptation routine) in Step (ii) could increase the numerical accuracy.

Step (iii):

[12]However, it turns out that for higher accuracy this method is not so efficient and is suggested to be replaced by a more efficient method in future research.

The last step is in principle straightforward, because we do now have all the necessary ingredients to compute the Sharpe ratio S and its upper bound S_B. However, since all the numerical values entering these computations are subject to numerical errors we have to take care about the numerical stability of the respective magnitudes. While the bound $S_B = \sigma(m(x))/E(m(x))$ for the Sharpe ratio turns out to be numerically nice, the first formula for the Sharpe ratio

$$\left|\frac{E(R(x)) - R^f(x)}{\sigma(R(x))}\right| \tag{4.10}$$

turns out to be considerably less precise than the second formula

$$\frac{-R^f(x)\mathrm{cov}\Big(m(x), R(x)\Big)}{\sigma(R(x))}. \tag{4.11}$$

Since the denominator is the same in both formulas, the difference can only be caused by the different numerators. A further investigation reveals that the numerator of the first formula can be rewritten as

$$R^f(x)(1 - E(m(x))E(R(x)))$$

while that of the second formula reads

$$R^f(x)(E(m(x)R(x)) - E(m(x))E(R(x))).$$

Note that in both formulas we have to subtract values which have approximately the same values, which considerably amplifies the numerical errors. As mentioned above, we know that $E(m(x)R(x)) = 1$, which shows that these formulas are theoretically equivalent. The higher accuracy of the second formula can be explained as follows: Assume that we have a small systematic additive numerical error in $R(x)$, e.g., $R_{num}(x) \approx R(x) + \delta$. Such errors are likely to be caused by the interpolation process on the grid. Then, using $R^f(x) = 1/E(m(x))$, in the first formula we obtain

$$\begin{aligned} R^f(x)(1 - E(m(x))E(R_{num}(x))) &\approx R^f(x)(1 - E(m(x))E(R(x) + \delta)) \\ &\approx R^f(x)(1 - E(m(x))E(R(x))) - \delta, \end{aligned}$$

while in the second formula we obtain

$$\begin{aligned} & R^f(x)(E(m(x)R_{num}(x)) - E(m(x))E(R_{num}(x))) \\ \approx\ & R^f(x)(E(m(x)(R(x) + \delta)) - E(m(x))E(R(x) + \delta)) \\ \approx\ & R^f(x)(E(m(x)R(x)) + E(m(x))\delta - E(m(x))E(R(x)) - E(m(x))\delta) \\ \approx\ & R^f(x)(E(m(x)(R(x))) - E(m(x))E(R(x))), \end{aligned}$$

i.e., systematic additive errors cancel out in the second formula.

5 The Stochastic Growth Model

Next we perform numerical computations for the stochastic growth model as present in Brock and Mirman (1972) and Brock (1979, 1982). We use a basic version as suggested and used by Santos nd Vigo-Aguiar (1998) which employs an aggregate capital stock and log utility.

The dynamics are defined by

$$\begin{aligned} k_{t+1} &= z_t A k_t^{\alpha} - c_t \\ \ln z_{t+1} &= \rho \ln z_t + \varepsilon_t \end{aligned}$$

where A, α and ρ are real constants and the ε_t are i.i.d. random variables with zero mean. The return function in (4.1) is $u(c) = \ln c$.

In our numerical computations we used $y_t = \ln z_t$ instead of z_t as the second variable. For our numerical experiments we employed the values

$$A = 5, \quad \alpha = 0.34, \quad \rho = 0.9, \quad \beta = 0.95,$$

and ε_t was chosen as a Gaussian distributed random variable with standard deviation $\sigma = 0.008$, which we restricted to the interval $[-0.032, 0.032]$. With these parameters it is easily seen that the computational domain $\Omega = [0.1, 10] \times [-0.32, 0.32]$ is controlled invariant, i.e., for any $(k, z) \in \Omega$ there exists c such that $\varphi((k, z), c, \varepsilon) \in \Omega$ for all values $\varepsilon \in [-0.032, 0.032]$. Therefore, this Ω can be used as our basis for the state space discretization.

The following figures show some of the numerically obtained results for this example. Figure 5.1 shows the optimal value function, the resulting final adaptive grid with 2977 nodes, and the corresponding asset price p. Figure 5.2 shows the k–components of several optimal trajectories with different initial values. Observe that all trajectories end up near the "equilibrium" point $(k, \ln z) = (2, 0)$ around which they oscillate due to the stochastic influence.

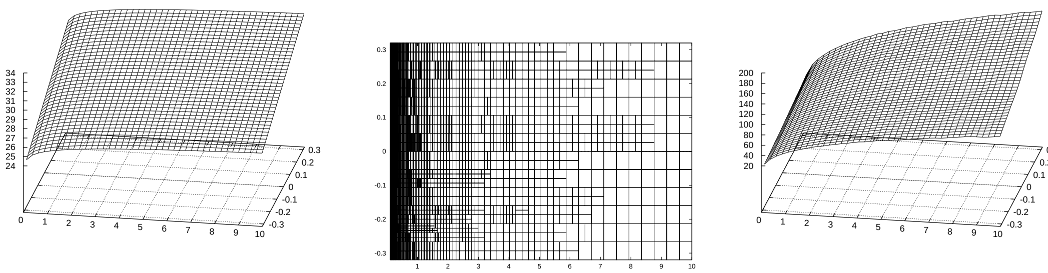

Figure 5.1: Optimal value function V, final adaptive grid and asset price function p

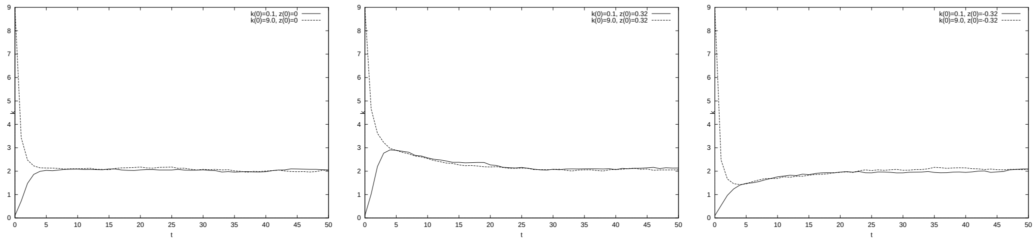

Figure 5.2: k–components of optimal trajectories for different initial values

For our stochastic growth model the optimal control is known analytically and is given by

$$c(x) = c(k, z) = (1 - \alpha\beta)Azk_t^\alpha.$$

Note that c here depends linearly on z.

Since $u'(c) = 1/c$, the stochastic discount factor becomes

$$m(x_t) = \beta\frac{c(x_t)}{c(x_{t+1})}.$$

Furthermore, in this model the dividend is given by $d(x) = c(x)$, and from this one easily computes[13]

$$p(x) = \frac{\beta}{1-\beta}c(x) = \frac{\beta}{1-\beta}(1 - \alpha\beta)Azk_t^\alpha.$$

Using these values, we can compute both the Sharpe ratio and its upper bound by evaluating the respective expressions numerically. We obtain the values

$$S = 0.007999 \text{ and } S_B = 0.102976.$$

Both values are indeed independent of x.

It should, however, be noted that the component parts defining S and S_B are not constant at all. In Figure 5.3 the values $R^f(x) = 1/E(m(x))$, $\sigma(m(x))$ and $-R^f(x)\text{cov}(m(x), R(x))$, defining the numerator of the Sharpe ratio, are shown depending on x, all computed from the exact values.

[13]see also Brock (1982).

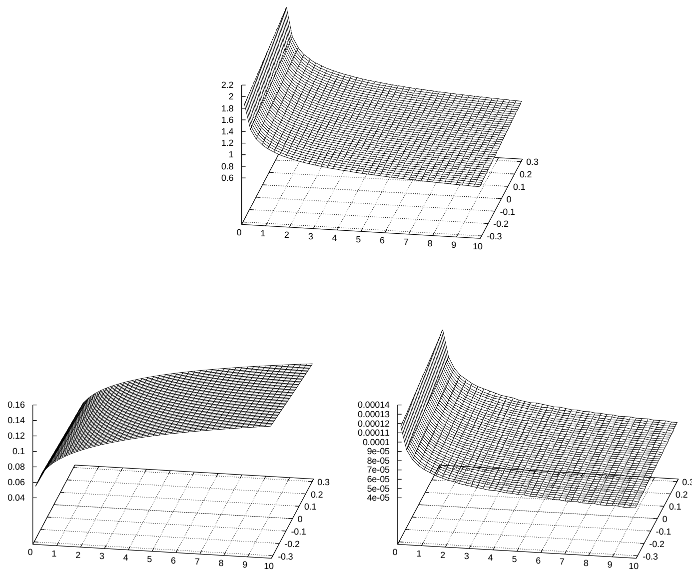

Figure 5.3: Functions $R^f(x)$, $\sigma(m(x))$ and $-R^f(x)\text{cov}\Big(m(x), R(x)\Big)$

In addition, Figure 5.4 shows a larger plot of the risk free interest rate $R^f(x)$ together with a plane at level $R^f = 1$, which allows to identify where the rate is below and where above one. Note that in the equilibrium $x = (k, \ln z) = (2, 0)$, around which the optimal trajectories end up in the long run, the value of R^f is 1.06.

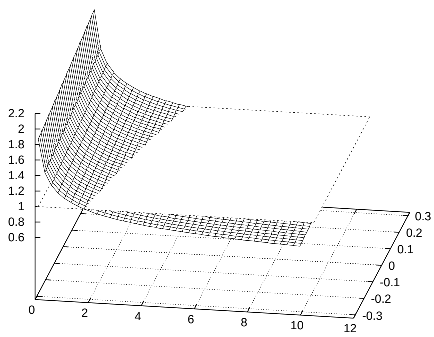

Figure 5.4: The risk free interest rate $R^f(x)$ and a plane at level $R^f = 1$

For our numerical tests we used two different accuracies. For a test with lower accuracy we used a discretization of the set C of control values with discretization step size $6.25 \cdot 10^{-2}$ and an adaptive grid with 2977 nodes. These computations were also used for the Figures

5.1 and 5.2 of the numerical examples (visually the results with higher accuracy look exactly the same).

For the test with higher accuracy we used a discretization of C with step size $2 \cdot 10^{-3}$ and an adaptive grid with 8108 nodes. Table 5.1 shows the respective results.

accuracy	error in V	error in c	error in p	error in S	error in S_B
coarse	$4.3 \cdot 10^{-3}$	$8.3 \cdot 10^{-2}$	4.072	$6.9 \cdot 10^{-3}$	$5.4 \cdot 10^{-4}$
fine	$1.3 \cdot 10^{-3}$	$5.4 \cdot 10^{-2}$	2.137	$4.7 \cdot 10^{-3}$	$3.9 \cdot 10^{-4}$

Table 5.1: Approximation errors for the fully numerical approach

The error in $\sigma(m)$ and in R^f were also computed and turned out to be of the magnitude of 10^{-3} and 10^{-2}, respectively.

As an alternative, we used the same grids but using the exact $c(x)$ in all our computations (i.e., for p, S and S_B). The corresponding accuracies are summarized in Table 5.2.

accuracy	error in V	error in c	error in p	error in S	error in S_B
coarse	–	0.0	$1.4 \cdot 10^{-1}$	$6.0 \cdot 10^{-6}$	$< 10^{-6}$
fine	–	0.0	$4.9 \cdot 10^{-2}$	$< 10^{-6}$	$< 10^{-6}$

Table 5.2: Approximation errors using analytically determined (i.e. exact) c

From the tables it is interesting to see how the original numerical errors in V, c and p propagate through the consecutive numerical approximations. In particular, it turns out that the error in S and S_B is much smaller than the error in p and slightly smaller than the error in c; in fact in all examples we investigated it is very similar to the error in the underlying optimal value function V from which c is obtained.

We also see that for the exact c even on the coarser grid very good approximations for S are obtained. From our experiments it seems that the crucial point in the numerical procedure is the numerical accuracy for the underlying V and c, while the accuracy of p does not seem to be crucial. Summarizing, if the consumption strategy c is either known exactly or if it can be determined numerically with a sufficiently high accuracy, then our numerical approach can be expected to be successful.

Note that we discuss absolute errors here. The reason for this lies in the fact that in our examples the stochastic process determining the magnitude of the Sharpe ratio does not seem to affect the accuracy, as we will demonstrate in our next numerical test.

We should mention that in this examples for $\sigma = 0.008$, for the fully numerical approach the errors for the computation of the Sharpe ratio S are of the same order as the Sharpe ratio itself which means that they do not provide a reasonable approximation. In other words, the relative error for S is of the order of one, i.e. the numerical results could be considered useless, Yet, we would like to see whether this undesirable behavior remains true if we modify our model parameters in such a way that the Sharpe ratio (which is unrealistically small as compared to the empirical Sharpe ratio[14]) increases.

To this end we changed our random variables ε_i to a Gaussian distributed random variable with standard deviation[15] $\sigma = 0.018$, which we now restricted to the interval $[-0.072, 0.072]$. In order not to increase the computational domain Ω we reduced the parameter ρ to $\rho = 0.5$ which ensures that $\Omega = [0.1, 10] \times [-0.32, 0.32]$ remains controlled invariant.

The exact values of S and S_B for these parameters evaluate to

$$S = 0.017994 \text{ and } S_B = 0.104366.$$

For this set of parameters we have repeated the fully numerical computation on an adaptive grid with 2624 nodes, which is roughly the same amount of nodes as for the coarse accuracy, above. Table 5.3 shows the respective numerical accuracies for $\sigma = 0.018$ compared to the accuracies for the model with lower standard deviation $\sigma = 0.008$ from Table 5.1.

σ	error in V	error in c	error in p	error in S	error in S_B
0.008	$4.3 \cdot 10^{-3}$	$8.3 \cdot 10^{-2}$	4.072	$6.9 \cdot 10^{-3}$	$5.4 \cdot 10^{-4}$
0.018	$3.7 \cdot 10^{-3}$	$8.1 \cdot 10^{-2}$	2.538	$5.5 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$

Table 5.3: Approximation for fully numerical approach and different σ

It turns out that the absolute numerical error in the approximation S does not increase when σ increases, implying that the numerical accuracy of our procedure appears to be independent of the standard deviation σ of the underlying random process and thus of the magnitude of the Sharpe ratio. In other words, when S increases then the relative error in S can be expected to decrease.

6 Conclusions

In recent times extensive research effort has been devoted to study the asset price characteristics, such as the risk-free interest rate, the equity premium and the Sharpe ratio, based on the stochastic growth model originating in the work by Brock. The failure of the model to match the empirical characteristics of asset prices and returns has given rise to numerous extensions of the basic model that allows for different preferences and technology shocks, adjustment cost of capital, two sector economies and heterogenous firms[16] and households. The aim of our paper was not to resolve the asset market puzzles along

[14]For reports on empirically realistic Sharpe ratios, see for example, Boldrin, Christiano and Fisher (2001).

[15]The standard deviation of $\sigma = 0.018$ is suggested in the RBC literature, see Boldrin, Christiano and Fisher (2001).

[16]A model with heterogenous firms in the context of a Brock type stochastic growth model can be found in Akdeniz and Dechert (1997) who are able to match, to some extent, the equity premium by building on idiosynchratic stochastic shocks to firms.

those lines of extension but rather to introduce and apply a method, a stochastic dynamic programming algorithm, that provides global and accurate solutions and that can easily be applied to those extended versions of the basic stochastic growth model. In order to test our algorithm we apply it to a basic stochastic growth model for which consumption, asset prices and the Sharpe ratio can analytically be computed. Overall, as we show, the errors depend on the discretization step size and grid refinement. Our computations, however, show that the optimal consumption, the value function and the Sharpe ratio can be computed with small absolute errors, even if larger errors for the asset prices arise. We also can observe, although the component parts of the Sharpe ratio depend on the state variables, the Sharpe ratio itself, as a measure of excess return per unit of risk, is a constant. Moreover, we want to stress that although the relative error in computing the Sharpe ratio can be of the order of one if the shocks, and thus the Sharpe ratio, are small, our numerical results show that the error in the computation of the Sharpe ratio is independent of the standard deviation of the underlying random process and thus of the magnitude of the Sharpe ratio. In other words the relative error decreases with increasing Sharpe ratio.[17] Thus, overall the results presented here are very encouraging and our method can be suggested to be applied to extended versions of the stochastic growth model.

[17]Note that in empirical research a Sharpe ratio is obtained which is by a factor of 40 to 50 greater than our smallest Sharpe ratio, the Sharpe ratio resulting from $\sigma = 0.008$, see Boldrin, Christiano and Fisher (2001).

References

[1] Akdeniz, L. and W.D. Dechert (1997), Do CAPM results hold in a dynamic economy? Journal of Economic Dynamics and Control 21: 981-1003.

[2] Breeden, D.T. (1979), An intertemporal asset pricing model with stochastic consumption and investment opportunities. Journal of Financial Economics 7: 231-262.

[3] Boldrin, M., L.J. Christiano and J.D.M. Fisher (2001), Habit persistence, asset returns and the business cycle. American Economic Review, vol. 91, 1: 149-166.

[4] Brock, W. (1979) An integration of stochstic growth theory and theory of finance, part I: the growth model, in: J. Green and J. Schenkman (eds.), New York, Academic Press: 165-190.

[5] Brock, W. (1982) Asset pricing in a production economy, in: The Economies of Information and Uncertainty, ed. by J.J. McCall, Chicgo, University pf Chicago Press: 165-192.

[6] Brock, W. and L. Mirman (1972), Optimal economic growth and uncertainty: the discounted case, Journal of Economic Theory 4: 479-513.

[7] Burnside, C. Discrete state–space methods for the study of dynamic economies, in: Marimon, R. and A. Scott, eds., Computational Methods for the Study of Dynamic Economies, Oxford University Press, 95–113, 2001.

[8] Candler, G.V. Finite–difference methods for continuous–time dynamic programming, in: Marimon, R. and A. Scott, eds., Computational Methods for the Study of Dynamic Economies, Oxford University Press, 172–194, 2001.

[9] Campbell, J.Y. and J.H. Cochrane (1999), Explaining the Poor Performance of Consumption-based Asset Pricing Models, Working paper, Harvard University.

[10] Chow, C.-S. and J.N. Tsitsiklis. An optimal one–way multigrid algorithm for discrete–time stochastic control. IEEE Trans. Autom. Control 36:898–914, 1991.

[11] Cochrane, J. (2001), Asset pricing, Princeton University Press, Princeton.

[12] Daniel, J.W. Splines and efficiency in dynamic programming, J. Math. Anal. Appl. 54:402–407, 1976.

[13] Falcone, M. (1987), A numerical approach to the infinite horizon problem of deterministic control theory, Appl. Math. Optim., 15: 1–13

[14] Falcone, M. and R. Ferretti, Convergence analysis for a class of high-order semi-Lagrangian advection schemes. SIAM J. Numer. Anal. 35:909–940, 1998.

[15] Grüne, L. (1997), An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation, Numer. Math., 75: 1288-1314.

[16] L. Grüne (2003), Errorr estimation and adaptive discretization for the discrete stochastic Hamilton–Jacobi–Bellman equation. Preprint, University of Bayreuth. Submitted, `http://www.uni-bayreuth.de/departments/math/~lgruene/papers/`.

[17] L. Grüne and W. Semmler (2003), Using dynamic programming with adaptive grid scheme for optimal control problems in economics. Working Paper No. 38, Center for Empirical Macroeconomics, University of Bielefeld. Forthcoming, Journal of Economic Dynamics and Control.

[18] Hansen, L.P. and T. Sargent (2001) Robust control, book manuscript, Stanford Unviersity.

[19] Jerman, U.J. (1998), Asset pricing in production economies, Journal of Monetary Economies 41: 257-275.

[20] Johnson, S.A., J.R. Stedinger, C.A. Shoemaker, Y. Li, J.A. Tejada–Guibert, Numerical solution of continuous–state dynamic programs using linear and spline interpolation, Oper. Research 41:484–500, 1993.

[21] Judd, K.L. Approximation, perturbation, and projection methods in economic analysis, Chapter 12 in: Amman, H.M., D.A. Kendrick and J. Rust, eds., Handbook of Computational Economics, Elsevier, pp. 511–585, 1996.

[22] Judd, K.L. and S.-M. Guu. Asymptotic methods for aggregate growth models. Journal of Economic Dynamics & Control 21: 1025-1042, 1997.

[23] Keane, M.P. and K.I. Wolpin. The Solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte Carlo evidence, The Review of Economics & Statistics, 76:648–672, 1994.

[24] Lettau, M. and H. Uhlig (1999), Volatility bounds and preferences: an analytical approach, revised from CEPR Discussion Paper No. 1678.

[25] Lettau, M. G. Gong and W. Semmler (2001), Statistical estimation and moment evaluation of a stochastic growth model with asset market restrictions, Journal of Economic Organization and Behavior, Vol. 44: 85-103.

[26] Lucas, R. Jr. (1978), Asset prices in an exchange economy. Econometrica 46: 1429-1446.

[27] Marcet, A. (1994) Simulation analysis of stochastic dynamic models: applications to theory and estimation, in: C.A. Sims, ed., Advances in Econometrics, Sixth World Congress of the Econometric Society, Cambridge University Press, pp. 81–118.

[28] McGrattan, E.R. and E.C. Prescott (2001) Taxes, regulation and asset prices. Working paper, Federal Reserve Bank of Minneapolis.

[29] Munos, R. and A. Moore. Variable resolution discretization in optimal control, Machine Learning 49:291–323, 2002.

[30] Reiter, M. Solving higher–dimensional continuous–time stochastic control problems by value function regression, J. Econ. Dyn. Control, 23:1329–1353, 1999.

[31] Rouwenhorst, K.G. (1995), Asset pricing implications of equilibrium business cycle models, in: T. Cooley: Frontiers of Business Cycle Research. Princeton, Princeton University Press: 295-330.

[32] Rust, J. Numerical dynamic programming in economics, in: Amman, H.M., D.A. Kendrick and J. Rust, eds., Handbook of Computational Economics, Elsevier, pp. 620–729, 1996.

[33] Rust. J. Using randomization to break the curse of dimensionality, Econometrica 65:478–516, 1997.

[34] Santos, M.S. and J. Vigo–Aguiar (1998), Analysis of a numerical dynamic programming algorithm applied to economic models, Econometrica 66: 409-426.

[35] Semmler, W. (2003), Asset prices, booms and recessions, Springer Publishing House, Heidelberg and New York.

[36] Tauchen, G. and R. Hussey. Quadrature–based methods for obtaining approximate solutions to nonlinear asset–price models, Econometrica, 59:371–396, 1991.

[37] Taylor J.B. and H. Uhlig. Solving non-linear stochastic growth models: a comparison of alternative solution methods. Journal of Business and Economic Studies, 8: 1-18, 1990.

[38] Trick, M.A. and S.E. Zin. A linear programming approach to solving stochastic dynamic programs, Working Paper, Carnegie–Mellon University, 1993.

[39] Trick, M.A. and S.E. Zin. Spline approximations to value functions: a linear programming approach, Macroeconomic Dynamics, 1:255-277, 1997