

# Using Dynamic Programming with Adaptive Grid Scheme for Optimal Control Problems in Economics\*

Lars Grüne

Mathematisches Institut, Universität Bayreuth<sup>†</sup>

Willi Semmler

CEM, Bielefeld and New School University, New York

**Abstract:** The study of the solutions of dynamic models with optimizing agents has often been limited by a lack of available analytical techniques to explicitly find the global solution paths. On the other hand the application of numerical techniques such as dynamic programming (DP) to find the solution in interesting regions of the state space was restricted by the use of fixed grid size techniques. Following Grüne (1997) and (2003) in this paper an adaptive grid scheme is used for finding the global solutions of discrete time Hamilton-Jacobi-Bellman (HJB) equations. Local error estimates are established and an adapting iteration for the discretization of the state space is developed. The advantage of the use of adaptive grid scheme is demonstrated by computing the solutions of one and two dimensional economic models which exhibit steep curvature, complicated dynamics due to multiple equilibria, thresholds (Skiba sets) separating domains of attraction and periodic solutions. We consider deterministic and stochastic model variants. The studied examples are from economic growth, investment theory, environmental and resource economics.

**Keywords:** dynamic optimization, dynamic programming, adaptive grid scheme

**JEL Classification:** C61, C63

---

\*Previous versions of this paper have been presented at an ASSA-conference, a North-American meeting of the Econometric Society, at the SCE conference in Aix-en-Provence and at a workshop on Economic Dynamics in Leiden (2002). For comments we want to thank the participants of those conferences. We also want to thank Buz Brock and Michael Woodford for comments on previous versions of the paper. We are also grateful for suggestions from two referees of the journal.

<sup>†</sup>95440 Bayreuth, Germany, [lars.gruene@uni-bayreuth.de](mailto:lars.gruene@uni-bayreuth.de), Phone: +49-921-55-3281, Fax: +49-921-55-5361

## 1 Introduction

In recent times the lack of closed form solutions of dynamic models with optimizing agents has led to the use of computational methods to solve those models. Closed form solutions are available for special cases such as the linear-quadratic control problem, see Ljungqvist and Sargent (2001, ch. 4) and growth models with log-utility. While those special models suffice for the analytical study of a variety of economic problems in more general cases they are not sufficient for a robust analysis of interesting economic problems. For the more general cases numerical methods have been proposed in the literature. A detailed discussion of a variety of numerical methods and a comparison with our proposed method is provided in Section 3 below.

Our paper is concerned with a family of continuous and discrete time dynamic models with optimizing agents whose solutions are most conveniently approached by the method of dynamic programming. Dynamic programming provides the value function and the control variable in feedback form. Yet, the application of numerical methods such as dynamic programming to find the global dynamics in interesting regions of the state space were restricted by the use of fixed grid size techniques. Following Grüne (1997) in this paper an adaptive grid scheme is used for finding global solutions of models with dynamic optimization. As Santos and Vigo-Aguiar (1998), we use numerical value function iteration but we employ local error estimates based on a flexible grid scheme. Since those numerical methods provide us with approximate solutions only, it is essential to have accuracy estimates for the numerical methods employed.

We consider discounted continuous time and discrete time optimal control problems and flexible grid scheme based on local error estimates. The advantage of the use of flexible grid scheme is demonstrated by computing the value function and the control variable in feedback form of one and two dimensional economic models. In order to study the accuracy of our numerical methods when applied to economic models we first want to test our algorithm by studying a basic growth model of Brock and Mirman (1972) type for which the exact solution is known. This allows us to judge the accuracy of our numerical method and to explore whether the flexible grid scheme is accurately capturing important solution properties, for example, steep curvature of the value function. Our algorithm will be applied to a deterministic as well as to a stochastic version of the Brock and Mirman (1972) growth model.

Moreover in the economic literature there exist more complicated dynamic models with optimizing agents which have been a challenge to commonly used numerical techniques. Those models exhibit more complicated dynamics due to the existence of multiple steady state equilibria,<sup>1</sup> thresholds (Skiba-sets) separating domains of attraction and periodic solutions as attractors. Examples of such models can be found in the literature on economic growth and development.<sup>2</sup> Multiple steady states and thresholds can also arise in the

---

<sup>1</sup>If there are local attractors among the equilibria some authors characterize them as indeterminate equilibria. For a survey of models with indeterminacy, see Benhabib and Farmer (1999).

<sup>2</sup>In the latter type of models a convex-concave production function arises which leads to thresholds separating paths to low per capita income (poor) countries and high per capita income (rich) countries, see Skiba (1978) and Azariadis and Drazen (1990). In more recent growth models of the Lucas or Romer type multiple steady state growth paths may arise due to externalities and complementarities of inputs, or monopolistic competition, see for example, Benhabib, Perli and Xie (1994) and Benhabib and Perli (1994).

dynamic decision problem of the firm, for example due to relative adjustment costs of investment,<sup>3</sup> in resource economics and in ecological management problems.<sup>4</sup> Our paper studies a prototype model from each of those areas and applies the proposed dynamic programming algorithm with adaptive grid scheme to find the global dynamics.

The remainder of the paper is organized as follows. Section 2 describes the basic discretization procedure and its strategy of adaptive grid refinement, while Section 3 compares our approach to other advanced dynamic programming schemes found in the literature. Section 4 solves one dimensional control problems. Here we study the basic growth model for which the exact solution is known and compare the accuracy of our numerical methods with previously used methods. We then study one dimensional dynamic optimization models with multiple domains of attraction and thresholds. Section 5 then presents results for two dimensional examples which also can give rise to thresholds and periodic solutions as attractors. Section 6 concludes the paper.

## 2 Numerical Dynamic Programming

In this section we describe a dynamic programming algorithm which enables us to compute optimal value functions as well as optimal trajectories of discounted optimal control problems of the below type (2.1)–(2.2) or (2.3)–(2.4).

After stating the problem formulation we will briefly describe the basic discretization procedure which goes back to Capuzzo Dolcetta (1983) and Falcone (1987) (see also Capuzzo Dolcetta and Falcone (1989) and Bardi and Capuzzo Dolcetta (1997), Appendix A), then discuss the adaptive gridding strategy from Grüne (1997) and, finally, give some hints on implementational aspects.

### 2.1 Problem Formulation

We consider discounted optimal control problems, either given in continuous time  $t \in \mathbb{R}_0^+$  by

$$V(x) = \max_{u \in \mathcal{U}} \int_0^\infty e^{-\delta t} g(x(t), u(t)) dt \quad (2.1)$$

where

$$\frac{d}{dt} x(t) = f(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n \quad (2.2)$$

or in discrete time  $t \in \mathbb{N}_0$  given by

$$V(x) = \max_{u \in \mathcal{U}} \sum_{t=0}^{\infty} \beta^t g(x(t), u(t)) \quad (2.3)$$

<sup>3</sup>See for example, Feichtinger et al. (2001) and Hartl et al. (2000).

<sup>4</sup>For the former, see Sieveking and Semmler (1997), for the latter, see Brock and Starret (1999).

where

$$x(t+1) = \varphi(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n \quad (2.4)$$

Our algorithm is easily extended to stochastic optimal control problems, see Section 2.5, below, for details.

## 2.2 Discretization

We start describing the basic discretization technique, which is done using a two step or semi-Lagrangian discretization, first in time and then in space. In the first step, which is only necessary for continuous time problems (2.1)–(2.2), the continuous time optimal control problem is replaced by a first order discrete time approximation given by

$$V_h(x) = \max_{u \in \mathcal{U}_d} J_h(x, u), \quad J_h(x, u) = h \sum_{i=0}^{\infty} \beta^i g(x_h(i), u_i) \quad (2.5)$$

where  $\beta = 1 - \delta h$  and  $x_h$  is defined by the discrete dynamics

$$x_h(0) = x, \quad x_h(i+1) = \varphi_h(x_h(i), u_i) := x_h(i) + hf(x_h(i), u_i) \quad (2.6)$$

and  $h > 0$  is the discretization time step. Here  $\mathcal{U}_d$  denotes the set of discrete control sequences  $u = (u_1, u_2, \dots)$  for  $u_i \in U$ . If the original problem is of type (2.3)–(2.4), then it is already in the form (2.5)–(2.6) with  $h = 1$ .

The optimal value function  $V_h$  is the unique solution of the discrete Hamilton–Jacobi–Bellman equation

$$V_h(x) = \max_{u \in U} \{hg(x, u) + \beta V_h(\varphi(x, u))\}. \quad (2.7)$$

If we define the dynamic programming operator  $T_h$  by

$$T_h(V_h)(x) = \max_{u \in U} \{hg(x, u) + \beta V_h(\varphi(x, u))\} \quad (2.8)$$

then  $V_h$  can be characterized as the unique solution of the fixed point equation

$$V_h(x) = T_h(V_h)(x) \text{ for all } x \in \mathbb{R}^n. \quad (2.9)$$

The second step of the algorithm now approximates the solution on a grid  $\Gamma$  covering a compact subset  $\Omega$  of the state space. Note that optimal control problems in economics are usually defined on a non-compact state space and the reduction to a compact domain  $\Omega$  is a nontrivial task. Typically, we pick a reasonable set  $\Omega$  and consider only those trajectories which remain in  $\Omega$  for all future times. Often it is possible to identify sets  $\Omega$  which are optimally invariant, i.e., which have the property that any optimal trajectory of the unconstrained problem which starts in  $\Omega$  remains in  $\Omega$  for all future times. In this case, the restriction to  $\Omega$  does not change the optimal value functions, otherwise one has to interpret more carefully the final results. In any case, we assume that for any point  $x \in \Omega$  there exists at least one control value  $u$  such that  $x + hf(x, u) \in \Omega$  holds.

Denoting the nodes of the grid  $\Gamma$  by  $x^i$ ,  $i = 1, \dots, P$ , we are now looking for an approximation  $V_h^\Gamma$  satisfying

$$V_h^\Gamma(x^i) = T_h(V_h^\Gamma)(x^i) \quad (2.10)$$

for all nodes  $x^i$  of the grid, where the value of  $V_h^\Gamma$  for points  $x$  which are not grid points (these are needed for the evaluation of  $T_h$ ) is determined by interpolation. In our implementation we use rectangular grids with bilinear interpolation in 2d and linear interpolation in 1d.

Note that an approximately optimal control law (in feedback form for the discrete dynamics) can be obtained from this approximation by taking the value  $u^*(x) = u$  for  $u$  realizing the maximum in (2.7), where  $V_h$  is replaced by  $V_h^\Gamma$ . This procedure in particular allows the numerical computation of approximately optimal trajectories.

For a rigorous convergence analysis of this discretization scheme we refer to Bardi and Capuzzo Dolcetta (1997) and Falcone and Giorgi (1999). Here we only remark that the scheme converges at least with order  $\gamma$ , where  $\gamma > 0$  depends on the continuity properties of  $V$  and  $V_h$ . In particular, we have  $\gamma = 1/2$  for continuous time problems and  $\gamma = 1$  for discrete time problems if  $V$  and  $V_h$  are Lipschitz continuous and  $\gamma = 2$  for discrete time problems if  $V_h$  is twice continuously differentiable.

### 2.3 Adaptive Grids

In order to distribute the nodes of the grid efficiently, we make use of a posteriori error estimation. The idea here lies in the fact, that we want to solve the equation  $V_h(x) = T_h(V_h)(x)$  for all  $x \in \Omega$  but we do only solve this equation for the nodes  $x^i$  of the grid. Therefore, for the estimation of the gridding error we estimate the residual of the dynamic programming operator, i.e., the difference between  $V_h^\Gamma(x)$  and  $T_h(V_h^\Gamma)(x)$  for points which are not nodes of the grid. Thus, for each cell  $C_l$  of the grid  $\Gamma$  we compute local error estimates

$$\eta_l := \max_{x \in C_l} |T_h(V_h^\Gamma)(x) - V_h^\Gamma(x)|$$

It was shown in Grüne (1997) that the error estimators  $\eta_l$  give upper and lower bounds for the real error. More precisely, one can prove the estimates

$$\frac{1}{2} \max_l \eta_l \leq \max_{x \in \Omega} |V_h(x) - V_h^\Gamma(x)| \leq \frac{1}{1 - \beta} \max_l \eta_l. \quad (2.11)$$

Thus these local error estimates fit into the usual theory for error estimates for PDEs and can be used as indicators for the following local refinement strategy of the grid  $\Gamma$ :

- (0) Pick an initial grid  $\Gamma_0$  and set  $i = 0$ . Fix a refinement parameter  $\theta \in (0, 1)$  and a tolerance  $rtol > 0$ .
- (1) Compute the solution  $V_h^{\Gamma_i}$  on  $\Gamma_i$ .
- (2) Evaluate the error estimates  $\eta_l$ . If  $\eta_l < rtol$  for all  $l$  then stop.
- (3) Refine all cells  $C_j$  with  $\eta_j \geq \theta \max_l \eta_l$ , set  $i = i + 1$  and go to (1).

As an alternative to specifying  $rtol$  one could also stop the refinement iteration when a prescribed maximum number of elements is reached, which is what we did in the numerical experiments in this paper. In all these experiments we used  $\theta = 0.1$ .

It should be noted that both the upper and the lower bound in (2.11) are necessary for an efficient and reliable adaptation routine: the upper bound ensures that small local errors indeed imply small global errors, which means that refining elements with large local errors — thus reducing the local error — will be the right strategy to reduce the global error (the error estimate is *reliable*). On the other hand, the lower bound ensures that large local errors do indeed indicate that there is something wrong with the numerical solution, which means that we do only refine if there really are large errors (the estimate is *efficient* in the sense that we do not make useless refinements). Note that the gap between the upper and lower bound becomes large if  $\beta \approx 1$ ; in this case numerical experience shows that the upper bound is likely to become too conservative in the sense that the estimated errors typically overestimate the real error. However, since the lower bound is independent of  $\beta$ , even in the case  $\beta \approx 1$  — which causes problems in many numerical schemes — the efficiency of the adaptive gridding method is maintained, i.e., we do not make unnecessary refinements.

In practice, the error estimates  $\eta_l$  cannot be evaluated exactly, because it involves maximization over infinitely many points in the element  $C_l$ . Instead, we use a number of test points  $x_1^t, \dots, x_m^t$  in each element and evaluate  $\tilde{\eta}_l = \max \eta(x_i^t)$ . Figure 2.1 shows the test points we have used in the 2d examples. Note that this procedure may underestimate the real error estimate on very large elements, thus it is advisable not to choose a too coarse initial grid  $\Gamma_0$ .

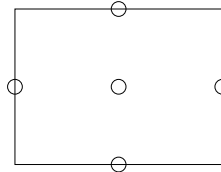


Figure 2.1: Test points  $x^t$  for a 2d element  $C_l$

This choice of test points naturally leads to the idea of anisotropic refinement of elements, i.e., the refinement of  $C_l$  not in all but only in selected coordinate directions, thus allowing to use “flat” or “stretched” elements. First note that for each rectangular element in  $\mathbb{R}^n$  we have several possibilities of refinement, as indicated in Figure 2.2 for the 2d case.



Figure 2.2: Refinements in 2d, from left to right: refinement in direction  $e_1$ ,  $e_2$ , and both

The simplest choice would be to refine each element in all possible coordinate directions,

like in the rightmost case in Figure 2.2. However, for an element  $C_l$  one can also consider the sets  $X_{new,i}$  of potential new nodes which would be added to  $\Gamma$  if the element  $C_l$  was refined in coordinate direction  $e_i$ . Figure 2.3 shows these points in 2d.

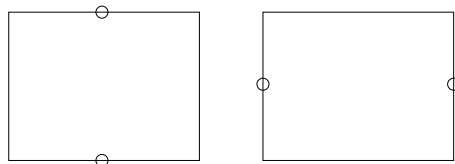


Figure 2.3: Potential new nodes  $X_{new,1}$  (left) and  $X_{new,2}$  (right) for a 2d element  $C_l$

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  $\eta_{dir} \leq \tilde{\eta}_l$  always holds. Since the points in  $X_{new,i}$  are contained in our selection of test points, anyway, we get these values for free. Now, if the error in one direction is large compared to the overall error  $\tilde{\eta}_l$ , then we only refine the element in the corresponding direction. We have used this strategy in all 2d examples presented in Section 5, but the effect is most apparent in Example 5.1, because here we have a strong curvature in  $e_1$ -direction but no curvature in  $e_2$ -direction.

During the adaptation routine it might happen that the error estimate on a coarser grid causes refinements in regions which after a few more refinement iterations 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  $V_{\Gamma_i}$  with its projection  $\tilde{V}_{\tilde{\Gamma}_i}$  onto a “thinned” grid  $\tilde{\Gamma}_i$  which is obtained from  $\Gamma_i$  by coarsening each element once. Using a specified coarsening tolerance  $ctol \geq 0$  one can add the following step after Step (2).

(2a) Coarsen all elements  $C_l$  with  $\eta_j < \theta \max_l \eta_l$  and  $\max_{x \in C_l} |V_{\Gamma_i}(x) - \tilde{V}_{\tilde{\Gamma}_i}(x)| \leq ctol$ .

This procedure also allows to start from rather fine initial grids  $\Gamma_0$ , which have the advantage of yielding a good approximation  $\tilde{\eta}_l$  of  $\eta_l$ . Unnecessarily fine elements in the initial grids will this way be coarsened afterwards. We have used this coarsening mechanism in the Examples 5.1 and 5.2, below, with its effect being most apparent in Example 5.2 where the final grid consists of fewer nodes than the initial grid.

## 2.4 Some Implementational Details

In this section we describe some implementational issues for our computations. We only outline the most important facts and refer to the appropriate literature for details. All numerical examples described in this paper were computed using an implementation of our algorithm in C on a Pentium III-PC with 997 MHz and 384 MByte RAM running under a SuSE LINUX system.

For the computation of  $V_h^\Gamma$  with time step  $h > 0$  on a grid  $\Gamma$  with  $P$  nodes  $x^1, \dots, x^P$ , we identify the piecewise (bi)linear function  $V_h^\Gamma$  with the  $p$ -dimensional vector  $V = (V_h^\Gamma(x^1), \dots, V_h^\Gamma(x^P))$ ,

$\dots, V_h^\Gamma(x^P))^T$  containing the values of  $V_h^\Gamma$  in the nodes  $x^i$  of the grid  $\Gamma$ . For the computation of  $V$  one could use the fixed point iteration

$$V^{j+1} = T_h(V^j),$$

starting, e.g., from the initial vector  $V^0 = (0, \dots, 0)^T$ . This procedure is known to converge to the right solution, however, the convergence can be rather slow. A first method to speed up the convergence is obtained if we replace the iteration  $V^{j+1} = T_h(V^j)$  by a Gauss–Seidel type iteration  $V^{j+1} = S(V^j)$  which is described in Grüne (1997, Sect. 3) under the name increasing coordinate algorithm.

In our implementation we use a straightforward discretization of  $U$  by a finite set  $\tilde{U}$ , for which the local maximization can be performed by simply comparing the finitely many values obtained for the finitely many control values from  $\tilde{U}$ . Note that other implementations use different strategies for this local optimization, e.g. in Santos and Vigo-Aguiar (1998) and Carlini, Falcone and Ferretti (2002) Brent’s Algorithm is used here; both references report good results for this optimization strategy. The advantage of using a finite set of control values  $\tilde{U}$  lies in the fact that one can easily couple this *value-space* iteration with a *policy-space* iteration as described in Gonzalez and Sagastizabal (1990) or Seek (1997): Once the control values in which the maxima for  $V^i$  and  $V^{i+1}$  are attained become convergent, we fix these values and solve the corresponding system of linear equations for these fixed control values using a fast iterative solver like the CGS or BICGSTAB method. After convergence of this method we continue with the usual iteration using  $S$  until the control values again converge, switch to a linear solver and so on. This combined policy–value–space iteration turns out to be much more efficient than the plain value space iteration using  $S$ , often more than 90 percent faster. We also take advantage of the fact that the solution on the grid  $\Gamma_i$  can be interpolated in the new nodes of  $\Gamma_{i+1}$  and then be used as the initial value for the iteration on  $\Gamma_{i+1}$ .

The adaptive rectangular grids were stored on a hierarchical tree data structure as described in Grüne, Metscher and Ohlberger (1999). In particular, this allows the compact storage of the complete adaptively refined grid without storing the coordinates of the nodes of the grid; these values are reconstructed when needed while traversing through the data structure. For a given point  $y \in \Omega$  this structure also allows a fast location of the grid element containing  $y$  which in turn leads to an efficient computation of the value  $V_h^\Gamma(\varphi(x, u))$  which is needed in the evaluation of the operator  $T_h$ .

## 2.5 Extension to stochastic optimal control problems

Our adaptive approach is easily extended to stochastic discrete time problems of the type

$$V(x) = E \left( \max_{u \in \mathcal{U}} \sum_{t=0}^{\infty} \beta^t g(x(t), u(t)) \right) \quad (2.12)$$

where

$$x(t+1) = f(x(t), u(t), z_t), \quad x(0) = x_0 \in \mathbb{R}^n \quad (2.13)$$

and the  $z_t$  are i.i.d. random variables. Again, this problem could be a priori in discrete time (in which case we have  $h = 1$ ) or it could be the time discretization of a continuous time stochastic optimal control problem with dynamics governed by an Itô stochastic differential equation, see Camilli and Falcone (1995).

The corresponding dynamic programming operator becomes

$$T_h(V_h)(x) = \max_{u \in U} \left\{ hg(x, u) + \beta E [V_h(\varphi(x, u, z))] \right\}, \quad (2.14)$$

where  $\varphi(x, u, z)$  is now a random variable.

If the random variable  $z$  is discrete then the evaluation of the expectation  $E$  is a simple summation, if  $z$  is a continuous random variable then we can compute  $E$  via a numerical quadrature formula for the approximation of the integral

$$\int_z V_h(\varphi(x, u, z))p(z)dz,$$

where  $p(z)$  is the probability density of  $z$ . Example 5.1 shows the application of our method to such a problem, where  $z$  is a truncated Gaussian random variable and the numerical integration was done via the trapezoidal rule.

It should be noted that despite the formal similarity, stochastic optimal control problems have several features different from deterministic ones. First, complicated dynamical behavior like multiple stable steady state equilibria, periodic attractors etc. is less likely because the influence of the stochastic term tends to “smear out” the dynamics in such a way that these phenomena disappear.<sup>5</sup> Furthermore, in stochastic problems the optimal value function typically has more regularity which allows the use of high order approximation techniques, see the following section for details. Finally, stochastic problems can often be formulated in terms of Markov decision problems with continuous transition probability (see Rust (1996) for details), whose structure gives rise to different approximation techniques, in particular allowing to avoid the discretization of the state space.

In these situations, our technique may not be the most efficient approach to these problems, and it has to compete with other efficient techniques, of which some are outlined in the following section. Nevertheless, Example 5.1 shows that adaptive grids are by far more efficient than non-adaptive methods if the same discretization technique (i.e., bilinear approximation in this example) is used for both approaches. It should also be noted that in the smooth case one can obtain estimates for the error in the approximation of the gradient of  $V_h$  from our error estimates, for details we refer to Grüne (2003).

### 3 Comparison with other dynamic programming schemes

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 computational technique more efficient. In this section

---

<sup>5</sup>A remark to this extent on an earlier version of the paper has been made by Buz Brock and Michael Woodford.

we will briefly review some of them and highlight the differences to our approach as described in the preceding section.

Before describing these different approaches, let us recall that the main purpose of our study is to compute the dynamics of the optimally controlled systems, in particular when complicated dynamic behavior like different domains of attractions separated by Skiba-Sets, periodic attractors and other phenomena occur. These phenomena occur in deterministic problems and their occurrence often comes along with a loss of regularity in the related value functions, which are at best Lipschitz but typically nonsmooth.

The majority of numerical dynamic programming approaches, however, do either need sufficient smoothness of the optimal value functions or stochastic problems with a sufficiently rich stochastic influence like in Markov decision problems with at least continuous transition probabilities. 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, see also the survey of Taylor and Uhlig (1990) where one can find a comparative numerical study of several methods.

Some of these methods (like Spline and piecewise high order approximation) use a grid discretization of the state space similar to our approach and it would be tempting to try whether adaptive discretization ideas based on our local error estimation technique work equally well with these approximation techniques; so far we are not aware of numerical studies in this direction. Again, we would like to recall that this approach needs appropriate smoothness which is not present in most of the examples we are interested in. Smoothness is also needed in other strategies, like in finite difference approximations (Candler (2001)), Gaussian Quadrature discretization (Tauchen and Hussey (1991), Burnside (2001)) and in perturbation techniques (Judd (1996)). While the last should also work if the value function is only piecewise smooth (which is the case in all of our examples), however, this technique computes the optimal value function starting from the stable equilibria and is typically less accurate away from these, which suggests that this may not be the best approach to detect Skiba points or curves, which are located far away from the stable equilibria, furthermore, the approach does not work if there are more complicated stable attractors like, e.g., periodic solutions as in Example 5.3, below.

A crucial aspect of any numerical dynamic programming method approximation is the curse of dimensionality, which states that the numerical effort (induced by the number of grid points) grows exponentially in the dimension of the state space (see Rust (1996) for a comprehensive account on complexity issues). Our adaptive approach will not be able to break the curse of dimensionality, but it helps to considerably reduce the computational cost for any fixed dimension. There are, however, methods that can break the curse of dimensionality by using either randomly distributed grid points (Rust (1997)) or so called low discrepancy grids (Rust (1996), Reiter (1999)). Even though these results are very appealing, these methods are unfortunately not applicable here because they either need Markov Decision Problems with (at least) continuous transition probability like in Rust (1996), which allows to avoid interpolation, or again sufficient smoothness of the optimal value function like in Reiter (1999) where interpolation is replaced by a regression technique. 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.

In this context it should be noted that from the complexity point of view for discretization techniques 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  $\Gamma_i$  is always used as the initial value for the computation on the next finer adaptive grid  $\Gamma_{i+1}$ . This also explains the large reduction in computation time observed for our approach compared to the computation on *one* fixed equidistant grid, see Tables 4.1 and 4.2, below.

Summarizing this first part of our comparative discussion, for smooth and “rich” stochastic problems a large amount of sophisticated and efficient techniques have been developed, which do, however, not apply to deterministic or nonsmooth problems. It would be very interesting to see how these methods compare to our adaptive approach in the smooth case, even more, it would be interesting to see whether the ideas of smooth approximation and adaptive discretization can be coupled in order to combine the strong sides of both approaches, but this is beyond the scope of this paper and will be the topic of future research.

As for deterministic and nonsmooth problems, there are still a number of advanced techniques reported in the literature which do all make use of adaptivity in one way or the other. 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 like (2.11) are not given. 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 analysing the information flow on the Markov chain related to the 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 Marcat (1994). This method, 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. Clearly, selecting a reasonable computational region is a serious problem at its own right, but for problems with multiple steady state equilibria and Skiba sets this approach is not adequate, because these sets have a very low probability (all trajectories move away) but are nevertheless very interesting for understanding the global dynamics of the problem. Furthermore, again this is a heuristic approach which needs to be complemented by a rigorous mathematical foundation.<sup>6</sup>

---

<sup>6</sup>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

Probably the adaptive approaches with the most solid mathematical background are presented in the papers of Trick and Zin (1993, 1997).<sup>7</sup> 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 upper bound in (2.11), 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 error estimate needs only one step of this trajectory.

Summarizing the second part of our discussion, there are a number of adaptive strategies around which are all reported to show good results, however, they are either heuristic and better suited for other classes of problems (like problems with discontinuities or problems with only one stable equilibrium), or they have nice theoretical features but are practically inconvenient because their implementation is numerically much more expensive than our approach.

Let us finally 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, however, to our knowledge it has not been used to control adaptive gridding strategies, and we are not aware of any estimates resembling (2.11), which is a crucial property for an efficient and reliable adaptive gridding scheme.

## 4 One Dimensional Examples

In this section we describe the application of our algorithm to a selection of one dimensional optimal control problems. For some one dimensional models the use of adaptive gridding strategies is not really necessary, because these problems can usually be solved with high

---

about the quality of the final solution of a heuristic method.

<sup>7</sup>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.

precision on fixed equidistant grids in a reasonable amount of time and with only low memory consumption. Nevertheless, even for these (numerically) simple problems one can show that adaptive gridding techniques can be much more efficient than algorithms on equidistant grids when there are kinks or steep curvature in the value function. Our first example will illustrate this, because for this problem the exact solution is known, hence the accuracy of the respective methods can be compared directly. For the subsequent examples the exact solution is not known, hence numerical methods are necessary for their analysis.

#### 4.1 A basic growth model with explicit solution

We start our numerical investigations with a basic growth model in discrete time, which goes back to Brock and Mirman (1972). This model has been used as a test example for many numerical algorithms, and our method can be regarded as an adaptive version of the numerical analysis performed by Santos and Vigo-Aguiar (1995 and 1998, Sect. 4). The problem is a discrete time maximization problem of type (2.3)–(2.4) with return function and dynamics given by

$$g(x, u) = \ln u \quad \text{and} \quad x(t+1) = Ax^\alpha - u(t),$$

with constants  $A > 0$  and  $0 < \alpha < 1$ . The exact solution to this problem is known (see Santos and Vigo-Aguiar (1998)) and is given by

$$V(x) = B + C \ln x, \quad \text{with } C = \frac{\alpha}{1 - \alpha\beta} \text{ and } B = \frac{\ln((1 - \alpha\beta)A) + \frac{\beta\alpha}{1 - \beta\alpha} \ln(\alpha\beta A)}{1 - \beta}.$$

For our numerical tests we specify  $A = 5$ ,  $\alpha = 0.34$  and the discount factor to  $\beta = 0.95$ . For these parameters the point  $x_1 = 2.06734$  is the unique optimal equilibrium. The corresponding optimal value function is shown in Figure 4.1.

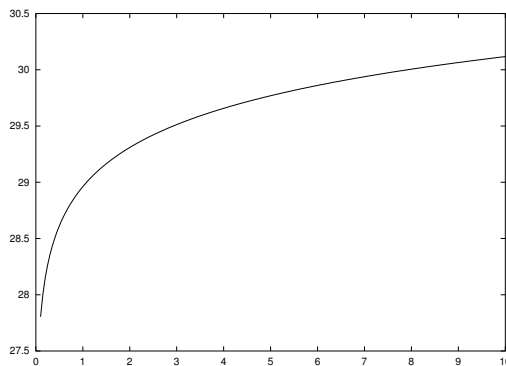


Figure 4.1: Optimal value function for Example 4.1

In Santos and Vigo-Aguiar (1998), this problem was solved with a similar algorithm as proposed here using equidistant grids. In our first computation we have also computed the solution on grids with an equidistant distribution of the nodes. We have computed the solution on the interval  $[0.1, 10]$  with control range  $U = [0.1, 5]$ , whose discretization  $\tilde{U}$

consisted of 501 equidistant values. Table 4.1 shows the respective results, the CPU time needed for the computation and also the maximal error (i.e., the error in the  $L_\infty$  norm) between the numerical and the exact solution. The achieved accuracy is comparable with the results from Santos and Vigo-Aguiar (1998) (small differences are due to the different local maximization strategy, see Sect. 2.4), the CPU times are, of course, not, because the computations here were performed with a different implementation on a different computer.

# nodes	Error	CPU time
99	$3.2 \cdot 10^{-2}$	1 s
989	$6.3 \cdot 10^{-4}$	9 s
9899	$5.0 \cdot 10^{-5}$	55 s

Table 4.1: Numerical solution of Example 4.1 on equidistant grids

Table 4.2 shows the respective results for the adaptive grid algorithm with refinement parameter  $\theta = 0.1$ . The CPU time in the last row is the overall time for the computation on all levels including all the time needed for the computation of the error estimates and the grid refinements. The estimated error is the error bound obtained from the upper inequality in (2.11), i.e.  $\max \eta_l / (1 - \beta)$ , in this example  $20 \cdot \max \eta_l$ .

# nodes	Error	estimated Error	CPU time
99	$3.2 \cdot 10^{-2}$	$5.9 \cdot 10^{-1}$	8 s
103	$1.3 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$	
110	$5.1 \cdot 10^{-3}$	$5.4 \cdot 10^{-2}$	
135	$2.6 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$	
183	$3.8 \cdot 10^{-4}$	$4.8 \cdot 10^{-3}$	
313	$1.1 \cdot 10^{-4}$	$1.2 \cdot 10^{-3}$	
595	$5.2 \cdot 10^{-5}$	$4.6 \cdot 10^{-4}$	

Table 4.2: Numerical solution of Example 4.1 on adaptive grid

It turns out that the final accuracy here is comparable to the equidistant grid with 9899 nodes, while the number of nodes needed here is only about 1/16 and the needed CPU time is only about 1/7 compared to the equidistant grid with same accuracy. In other words, in this example the use of the adaptive grid strategy yields a reduction of the memory consumption of about 94 per cent and a reduction of the computation time of about 85 per cent. Figure 4.2 shows the grid nodes of the adaptive grid depending on  $x$  where the mesh width is indicated by the  $y$ -position of the nodes (the lower, the finer). Note that due to the hierarchical construction of the grid the possible mesh widths are always of the form  $c2^{-k}$  for a fixed constant  $c > 0$  and  $k \in \mathbb{N}$ . Comparing this figure with the graph of the value function in Figure 4.1 one sees that the steeper the curvature of the graph of the optimal value function is, the finer the adaptive grid becomes.

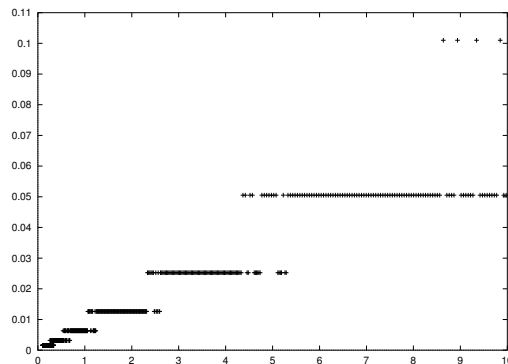


Figure 4.2: Mesh width of final adaptive grid for Example 4.1

## 4.2 A growth model with credit market

Next, we investigate a growth model from Grüne, Semmler and Sieveking (2001) which admits borrowing from the credit market for investment but where the maximal level of debt is given by the present value of an income stream from economic activity. Under the assumption of unrestricted access to capital market, a country, for example, may face a present value borrowing constraint which we want to compute. The interest rate paid on the debt is constant.<sup>8</sup> The model leads to an optimal control problem of type (2.1)–(2.2) with a return function and dynamics given by

$$g(x, u) = Ax^\alpha - u - u^\beta x^{-\gamma} \quad \text{and} \quad \frac{d}{dt}x(t) = u(t) - \sigma x(t),$$

where  $x$  denotes the capital stock,  $u$  is the investment and the discount rate  $\delta$  in (2.1) is the interest rate (for the numerical treatment of non-constant credit cost see Grüne, Semmler and Sieveking (2001)).

We have solved this problem with constants  $\sigma = 0.15$ ,  $A = 0.29$ ,  $\alpha = 1.1$ ,  $\beta = 2$ ,  $\gamma = 0.3$  and  $\delta = 0.1$ . The qualitative behavior of the numerical results, which were computed with  $h = 1/20$  and 401 control values in  $\tilde{U}$  covering the interval  $U = [0, 1/4]$ , is shown in the Figures 4.3–4.5. Unlike the previous example, given the parameter constellations of our model, this model exhibits multiple steady state equilibria. We obtain two optimal equilibria  $x_1 = 0$  and  $x_2 = 0.996$ , with domains of attraction separated by the Skiba point  $x^* = 0.267$ . Yet, as it is shown in Grüne, Semmler and Sieveking (2001) there are in fact three steady state equilibria, but the middle one in the vicinity of Skiba point  $x^* = 0.267$ , is unstable and non-optimal. The kink in the value function, however, is barely visible in Figure 4.3, but the jump in the control variable in Figure 4.4 and also the local mesh refinement (here the final grid consists of 116 nodes) clearly indicate the presence of a

<sup>8</sup>This model goes back to Blanchard (1983). In the original model by Blanchard, however, a utility functional is maximized. In Grüne, Semmler and Sieveking (2001) it is shown that this can be transformed into the below employed transformed model. Moreover, in Grüne, Semmler and Sieveking (2001) a variant with state dependent credit cost is treated which requires a second state equation. Our model can also be read as a growth model with poverty trap in the sense of Skiba (1978) and Azariadis and Drazen (1990), since the growth model exhibits a threshold below which per capita income will shrink.

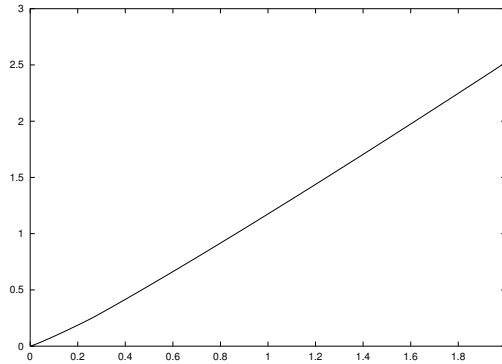


Figure 4.3: Optimal value function for Example 4.2

Skiba point which represents a threshold below which it is optimal to deplete the capital stock and above which the capital stock and thus per capita income will grow. Figure 4.5 shows the mesh width for the final adaptive grid of our computed example.

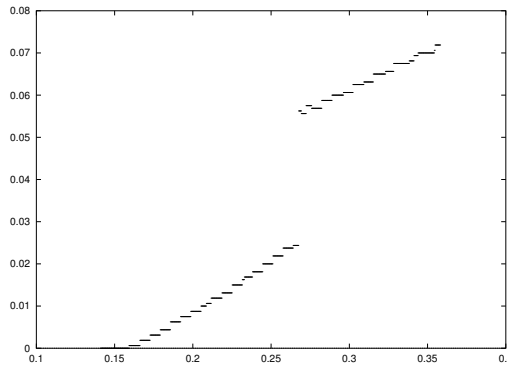


Figure 4.4: Optimal control for Example 4.2 in a neighborhood of the Skiba point

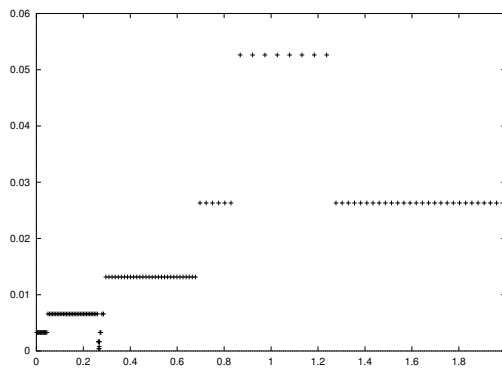


Figure 4.5: Mesh width of final adaptive grid for Example 4.2

Also in this example we have evaluated the error bound from inequality (2.11) which is given in Table 4.3.

# nodes	estimated Error
39	$6.1 \cdot 10^{-3}$
45	$4.8 \cdot 10^{-3}$
46	$3.0 \cdot 10^{-3}$
58	$4.2 \cdot 10^{-3}$
59	$1.4 \cdot 10^{-3}$
94	$2.6 \cdot 10^{-3}$
99	$1.8 \cdot 10^{-3}$
104	$1.0 \cdot 10^{-3}$
116	$4.0 \cdot 10^{-4}$

Table 4.3: Numbers of nodes and error bounds for Example 4.2

### 4.3 A 1d investment model with relative adjustment cost

Next, we investigate a continuous time decision problem of the firm where the firm's objective is to maximize, through the choice of an optimal investment strategy, the discounted value of its cash flows subject to some technology constraint. This type of decision problem which also exhibits multiple steady state equilibria may also result, as the example 4.2, in a discontinuous policy function and thus in a discontinuous investment strategy. A model of this type has been introduced, although not numerically solved, by Hartl et al. (2000). The return function and dynamics are given by

$$g(x, u) = ax - bx^2 - ux - cu - du^2 \quad \text{and} \quad \frac{d}{dt}x(t) = u(t)x(t) - \sigma x(t),$$

respectively, where  $x$  is the capital stock and  $ux$  is the investment (here we have performed the change of variables  $u = u/x$  on the model from Hartl et al. (2000) which removes the singularity of the return function  $g$  in  $x = 0$ ).

As the previous example, this model also exhibits multiple stable optimal equilibria for suitable parameters, whose domains of attraction are separated by a Skiba point, see Deissenberg et al. (2001) for a detailed explanation of this phenomenon. Here we have computed the optimal value function for  $\sigma = 0.1$ ,  $a = 1$ ,  $b = 0.6$ ,  $c = 0$ ,  $d = 0.3$  and discount rate  $\delta = 0.2$ . Figure 4.6 shows the corresponding optimal value function computed with time step  $h = 1/20$  and 501 equidistant control values in  $\tilde{U}$  discretizing the interval  $U = [0, 1]$ . With these parameters, the optimal equilibria are  $x_1 = 0$  and  $x_2 = 0.565$ . Yet, the Skiba point in between at  $x^* = 0.010$  is barely visible as a kink in the optimal value function in Figure 4.6. The Skiba point is clearly visible as a jump in the optimal feedback control law in Figure 4.7 where the optimal feedback law is shown.

Finally, the Skiba point also shows up in the mesh width of the final adaptive grid, which in this example consists of 399 nodes. The kink in the optimal value function produces large local error, which in turn result in a fine grid around this point. This effect is visible

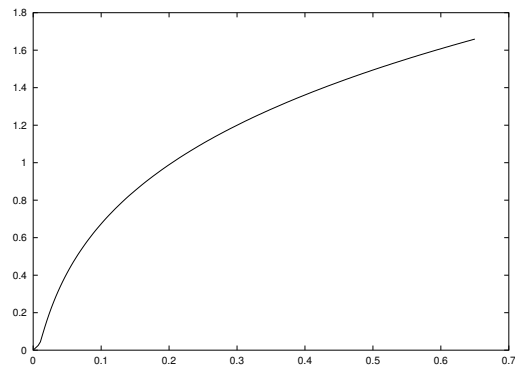


Figure 4.6: Optimal value function for Example 4.3

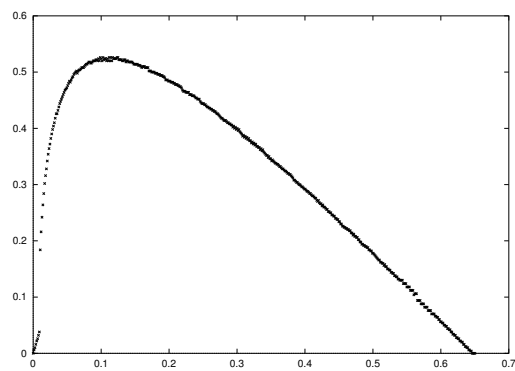


Figure 4.7: Optimal control for Example 4.3

in Figure 4.8. Note, however, that also the regions with steep curvature of the graph cause a fine resolution of the grid. Table 4.4 gives the estimated upper bound on the errors for the different adaptive grids.

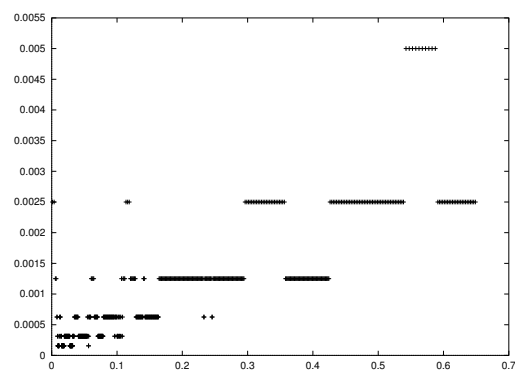


Figure 4.8: Mesh width of final adaptive grid for Example 4.3

# nodes	estimated Error
131	$2.1 \cdot 10^{-2}$
171	$1.1 \cdot 10^{-2}$
250	$5.7 \cdot 10^{-3}$
399	$5.3 \cdot 10^{-3}$

Table 4.4: Numbers of nodes and error bounds for Example 4.3

#### 4.4 An ecological management problem

Our final 1d example is an ecological management problem from Brock and Starret (1999). An extensive numerical study of this model was performed in Grüne, Kato and Semmler (2002). We here show a numerical computation with a specific parameter set.

The model is given by a return function and dynamics

$$g(x, u) = au^\sigma - k\beta x^2 \quad \text{and} \quad \frac{d}{dt}x(t) = u(t) - \mu x(t) + \frac{mx(t)^\rho}{n^\rho + x(t)^\rho}.$$

In the above the return function  $g(x, u)$  represents a utility function with two components: the utility of an affector, obtaining a benefit letting phosphorus,  $u$ , running into a lake and the disutility of an enjoyer of the lake being negatively affected by the stock of phosphorus,  $x$ . The state equation denotes the dynamics of the stock of phosphorus in the lake. We specify the parameters  $a = 2$ ,  $\sigma = \beta = \frac{1}{2}$ ,  $m = n = 1$ ,  $\rho = 2$ ,  $\mu = 0.55$  and discount rate  $\delta = 0.1$ . The numerical parameters were chosen as time step  $h = 1/20$  and 501 control values  $\tilde{U}$  discretizing  $U = [0, 0.4]$ . With these parameters, again multiple optimal equilibria  $x_1 = 0.396$  and  $x_2 = 1.861$  separated by a Skiba point  $x^* = 0.714$  arise, and consequently the Figures 4.9–4.11 show exactly the same qualitative behavior as in the previous two examples. The Skiba point is visible in all three figures: as a kink in the optimal value function Figure 4.9, as a jump in the optimal control (Figure 4.10) and refined elements in the adaptive grid (Figure 4.11). Table 4.5 again gives the corresponding upper error bounds from (2.11).

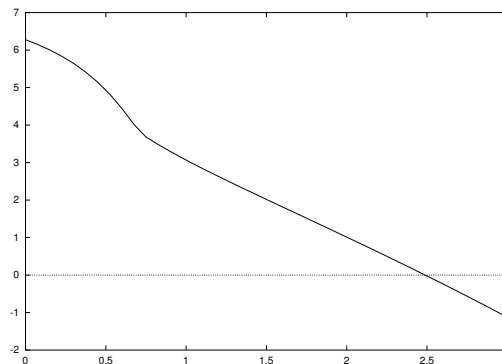


Figure 4.9: Optimal value function for Example 4.4

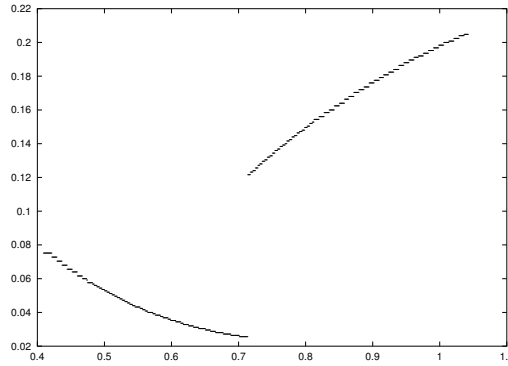


Figure 4.10: Optimal control for Example 4.4 in a neighborhood of the Skiba point

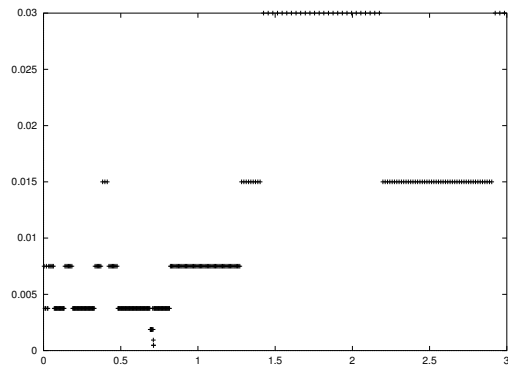


Figure 4.11: Mesh width of final adaptive grid for Example 4.4

## 5 Two Dimensional Examples

Next, we describe three two-dimensional problems which we have solved using our algorithm. We first study our above basic growth model of Sect. 4.1, but we employ an extension to a stochastic variant. This gives rise to two state variables. Next we turn to deterministic problems, where the study of the dynamics of two-dimensional problems with multiple steady state equilibria have been quite a challenge for research in economics

# nodes	estimated Error
101	$3.5 \cdot 10^{-1}$
122	$3.2 \cdot 10^{-1}$
133	$3.4 \cdot 10^{-1}$
135	$5.0 \cdot 10^{-2}$
225	$1.2 \cdot 10^{-1}$
240	$4.2 \cdot 10^{-2}$

Table 4.5: Numbers of nodes and error bounds for Example 4.4

since here one expects the separation of domains of attraction given not by threshold points but threshold lines (Skiba lines). Furthermore, two-dimensional models may give rise to periodic solutions as attractors. Examples of both cases are subsequently presented.

### 5.1 A 2d stochastic growth model

Our first 2d problem is a 2d version of the Brock and Mirman (1972) model from Example 4.1. Here the 1d model 4.1 is extended using a second variable modelling a stochastic shock. The model is given by the discrete time equations

$$\begin{aligned}x_1(t+1) &= x_2(t)Ax_1(t)^\alpha - u(t) \\x_2(t+1) &= \exp(\rho \ln x_2(t) + z_t)\end{aligned}$$

where  $A$ ,  $\alpha$  and  $\rho$  are real constants and the  $z(t)$  are i.i.d. random variables with zero mean. The return function in (2.3) is again  $g(x, u) = \ln u$ .

In our numerical computations we used the parameter values  $A = 5$ ,  $\alpha = 0.34$ ,  $\rho = 0.9$  and  $\beta = 0.95$ . As for Example 4.1, the exact solution is known and given by

$$V(x) = B + C \ln x_1 + Dx_2,$$

where

$$B = \frac{\ln((1 - \beta\alpha)A) + \frac{\beta\alpha}{1-\beta\alpha} \ln(\beta\alpha A)}{1 - \beta}, \quad C = \frac{\alpha}{1 - \alpha\beta}, \quad D = \frac{1}{(1 - \alpha\beta)(1 - \rho\beta)}$$

We have computed the solution to this problem on the domain  $\Omega = [0.1, 10] \times [-0.32, 0.32]$ . The integral over the Gaussian variable  $z$  was approximated by a trapezoidal rule with 11 discrete values equidistributed in the interval  $[-0.032, 0.032]$  which ensures  $\varphi(x, u, z) \in \Omega$  for  $x \in \Omega$  and suitable  $u \in U = [0.5, 10.5]$ . For evaluating the maximum in  $T$  the set  $U$  was discretized with 161 points. Table 5.1 shows the results of the resulting adaptive gridding scheme applied with refinement threshold  $\theta = 0.1$  and coarsening tolerance  $ctol = 0.001$ . Figure 5.1 shows the resulting optimal value function and adapted grid.

# nodes	Error	estimated Error
49	$1.4 \cdot 10^0$	$1.6 \cdot 10^{-1}$
56	$0.5 \cdot 10^{-1}$	$6.9 \cdot 10^0$
65	$2.9 \cdot 10^{-1}$	$3.4 \cdot 10^0$
109	$1.3 \cdot 10^{-1}$	$1.6 \cdot 10^0$
154	$5.5 \cdot 10^{-2}$	$6.8 \cdot 10^{-1}$
327	$2.2 \cdot 10^{-2}$	$2.4 \cdot 10^{-1}$
889	$9.6 \cdot 10^{-3}$	$7.3 \cdot 10^{-2}$
2977	$4.3 \cdot 10^{-3}$	$3.2 \cdot 10^{-2}$

Table 5.1: Number of nodes and errors for Example 5.1

In Santos and Vigo-Aguiar (1995) on equidistant grids with  $143 \times 9 = 1287$  and  $500 \times 33 = 16500$  nodes, errors of  $2.1 \cdot 10^{-1}$  and  $1.48 \cdot 10^{-2}$ , respectively, were reported. In our adaptive

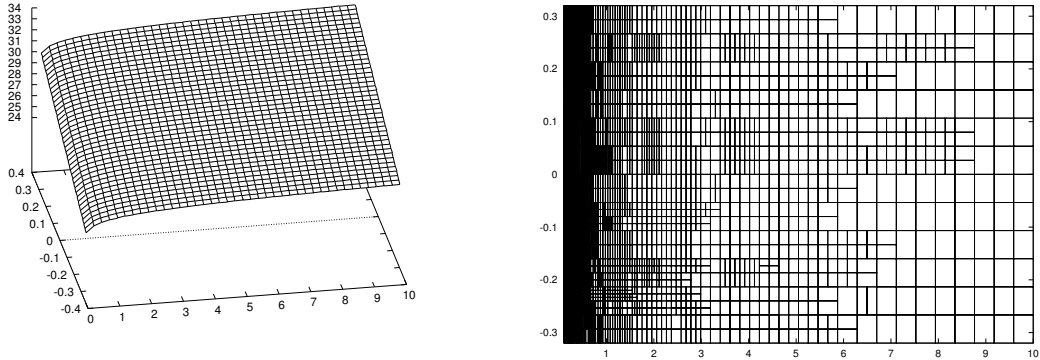


Figure 5.1: Approximated value function and final adaptive grid for Example 5.1

iteration these accuracies could be obtained with 109 and 889 nodes, respectively; thus we obtain a reduction in the number of nodes of more than 90% in the first and almost 95% in the second case, even though the anisotropy of the value function was already taken into account in these equidistant grids. Here again in our stochastic version of the growth model of Sect. 4.1, a large curvature in the value function can best be approximated with grid refinement.

## 5.2 A 2d investment model with relative adjustment cost

The following problem from Feichtinger et al. (2000) is a 2d variant of the 1d Example 4.3, where the control variable now is the change of investment rather than investment itself. It is given by

$$g(x, u) = k_1 \sqrt{x_1} - \frac{x_1}{1 + k_2 x_1^4} - c_1 x_2 - \frac{c_2 x_2^2}{2} - \frac{\alpha u^2}{2}$$

with dynamics

$$\frac{d}{dt} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} x_2(t) - \sigma x_1(t) \\ u(t) \end{pmatrix}.$$

Using the parameters  $\sigma = 0.25$ ,  $k_1 = 2$ ,  $k_2 = 0.0117$ ,  $c_1 = 0.75$ ,  $c_2 = 2.5$ ,  $\alpha = 12$  and discount rate  $\delta = 0.04$  we have obtained the results shown in the following Figures 5.2–5.3, with time step  $h = 1/20$  and 101 control values  $\tilde{U}$  covering  $U = [-1, 1]$ . Here we started from a rather fine grid and used the coarsening tolerance  $ctol = 0.001$ . As a result, the final grid is coarser than the initial grid but better adapted to the problem and thus provides a smaller error with fewer elements and grid nodes.

The final adaptive grid here consists of 14255 nodes. Note that the Skiba curve is visible in all three figures: as a discontinuity on the optimally controlled vector field (Figure 5.2, left), as a jump in the optimal control (Figure 5.2, right), as a kink in the optimal value function (Figure 5.3, left), and as a line of refined elements in the adaptive grid (Figure 5.3, right). This last figure in particular shows that the adaptive gridding strategy is very

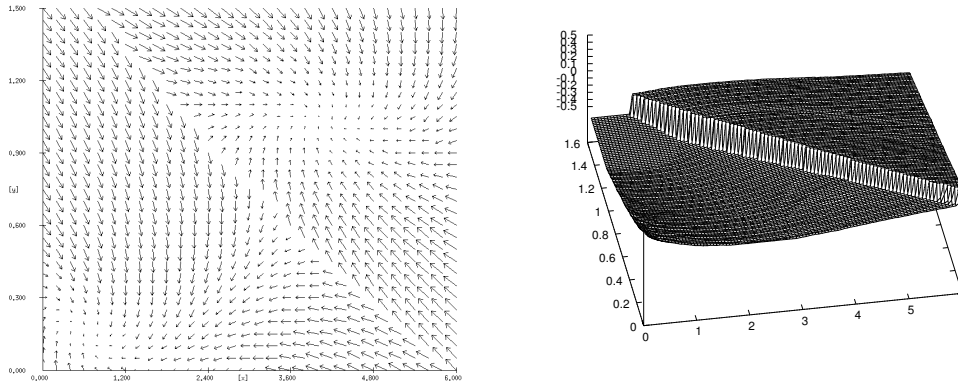


Figure 5.2: Optimal vector field and optimal control for Example 5.2

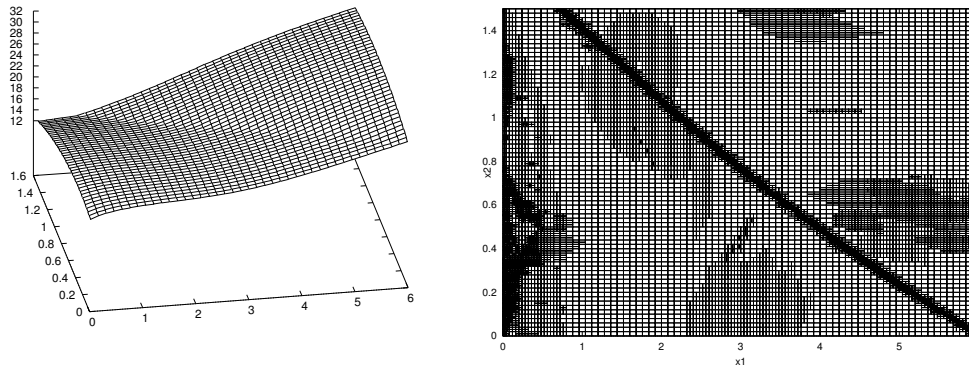


Figure 5.3: Optimal value function and final adaptive grid for Example 5.2

well suited for detecting threshold curves and provides an accurate resolution for such a model.

# nodes	estimated Error
22201	8.5
8264	6.8
9848	4.3
14255	3.6

Table 5.2: Numbers of nodes and error bounds for Example 5.2

Note that the estimated error values in Table 5.2 are very conservative here due to two reasons: first, the small denominator  $1 - \beta = 0.002$  in the right inequality of estimate (2.11) causes a large gap between the upper and lower bound for the error, and our numerical experience shows that the lower bound is often the more reliable one. Second, and more

important here, the large error values are concentrated along the Skiba curve, where large errors occur due to the fact that the position of the numerical curve is not precisely at its exact location. Away from this curve, the error estimates are much lower, and since trajectories do only move away from this unstable curve this means that also the error is much lower there. If we evaluate the error bound away from the Skiba curve we obtain an upper bound for the error of about  $10^{-1}$  and around the stable equilibria we obtain an even smaller bound of the order of  $10^{-3}$ .

### 5.3 A 2d model of interacting resources

As a last example we consider a model for the optimal exploitation of interacting resources from Semmler and Sieveking (1994) and analytically studied in Sieveking and Semmler (1997). It is given by the return function

$$g(x, u) = \frac{1}{1 + x_1 u} x_1 u + \frac{1}{1 + x_2 u} x_2 u - \frac{u}{2}$$

and the dynamics

$$\frac{d}{dt} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} (a_0 - a_2 x_2(t) - a_1 x_1(t) - u(t)) x_1(t) \\ (b_1 x_1(t) - b_0 - b_2 x_2(t) - u(t)) x_2(t) \end{pmatrix}.$$

Here  $x_1$  and  $x_2$  are the number of two species (e.g. fish in the ocean) with intraspecific competition, where  $x_2$  is the predator for the prey  $x_1$ . The control variable  $u$  models the harvesting effort chosen by the economic agent.

We set the parameters to  $a_0 = 1.04$ ,  $a_1 = 0.01$ ,  $a_2 = 0.07$ ,  $b_0 = 1.01$ ,  $b_1 = 0.2$ ,  $b_2 = 0.01$  and the discount rate to  $\delta = 5$ . We have computed the optimal value function on the domain  $\Omega = [0, 20] \times [0, 40]$ . The numerical time step was chosen as  $h = 1/20$  and the range of control values  $U = [0, 3]$  was discretized with 101 equidistributed values  $\tilde{U}$ . The Figures 5.4–5.5, below, show the results on  $[0, 20] \times [0, 30]$ , which is where the interesting dynamical behavior of the optimally controlled system occurs.

The final adaptive grid for this example consisted of 14208 nodes. The interesting fact in this system is that although the trajectories for constant harvesting effort do always tend to asymptotically stable equilibria, the optimal harvesting strategy here turns out to have an asymptotically stable periodic limit cycle. Figure 5.4 shows the vector field (left) the optimal control (right) and Figure 5.5 (left) the value function and the regions of grid refinement (right). Figure 5.6 shows this optimal periodic trajectory (as a solid line) along with two other optimal trajectories (as dotted lines) which tend to this periodic solution.

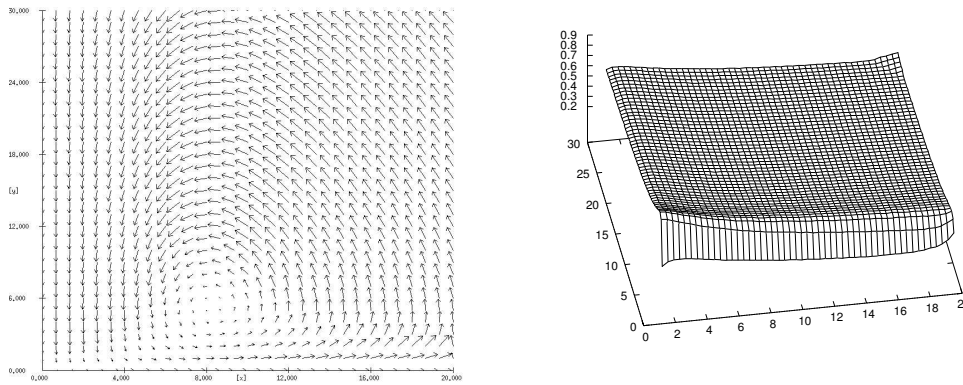


Figure 5.4: Optimal vector field and optimal control for Example 5.3

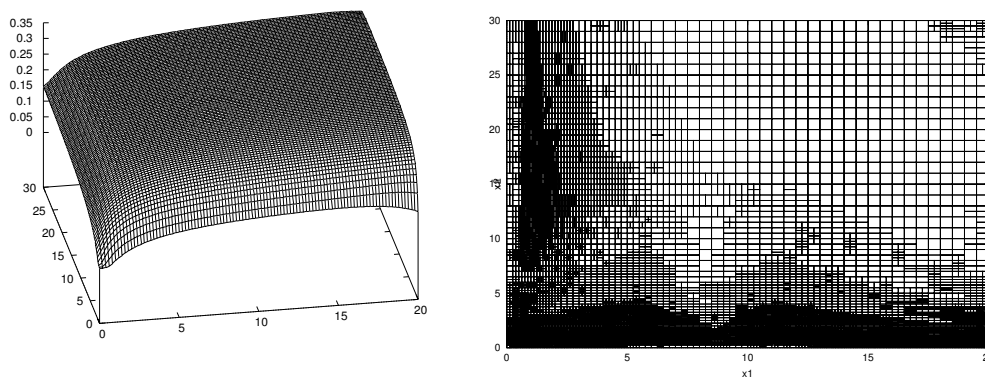


Figure 5.5: Optimal value function and final adaptive grid for Example 5.3

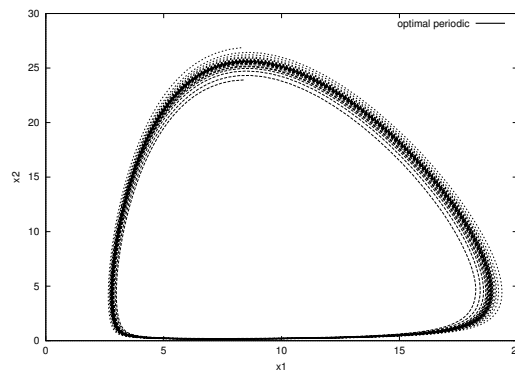


Figure 5.6: Optimal trajectories for Example 5.3

# nodes	estimated Error
1681	$2.6 \cdot 10^{-2}$
1827	$1.2 \cdot 10^{-2}$
2399	$6.9 \cdot 10^{-3}$
3544	$3.3 \cdot 10^{-3}$
6938	$1.7 \cdot 10^{-3}$
14208	$1.2 \cdot 10^{-3}$

Table 5.3: Numbers of nodes and error bounds for Example 5.3

## 6 Conclusions

The current paper uses dynamic programming with a flexible gridding strategy for the robust analysis of the dynamics of economic models with optimizing agents. The flexible gridding strategy permits to compute in models with multiple domains of attraction, threshold points (one dimensional examples) and threshold lines (two dimensional examples). The method is also well suited to accurately compute value functions with steep curvature or periodic solutions as attractors. Such solutions are computed in a robust way through local error analysis and grid refinements for prototype economic models of economic growth (deterministic or stochastic versions), dynamic decision problem of the firm and resource and ecological management problems. The global error estimate provided by the method turns out to yield a reliable upper bound for the real error, which, however, can be conservative for discount factor  $\beta \approx 1$ , in the sense that the real error is overestimated.

## References

- Azariadis, C. and A. Drazen, 1990. Threshold externalities in economic development. *Quarterly Journal of Economics*, 105, no. 2: 501-526.
- Bardi, M. and I. Capuzzo Dolcetta, 1997. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston.
- Benhabib, J., R. Perli and D. Xie, 1994. Monopolistic competition, indeterminacy and growth. *Ricerche Economiche*, 48: 279-298.
- Benhabib, J. and R. Perli, 1994. Uniqueness and indeterminacy: on the dynamics of endogenous growth. *Journal of Economic Theory*, 63, no. 1: 113-142.
- Benhabib, J. and R. Farmer, 1999. Indeterminacy in macroeconomics. In: J. Taylor and M. Woodford (eds.), *Handbook of Macroeconomics*, Elsevier publication, Amsterdam, New York.
- Blanchard, O.J, 1983. Debt and current account deficit in Brazil. In: P.A. Armella, R. Dornbush and M. Obstfeld, eds., *Financial Policies and the World Capital Market: the Problem of Latin American Countries*. Chicago: University of Chicago Press: 187-197.
- Brock, W. and L. Mirman, 1972. Optimal economic growth and uncertainty: the discounted case, *Journal of Economic Theory* 4:479–513.
- Brock, W.A. and D. Starrett, 1999. Nonconvexities in ecological management problems. SSRI Working Paper 2026, Department of Economics, University of Wisconsin – Madison.
- Burnside, C., 2001. 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.
- Camilli, F. and M. Falcone, 1995. An approximation scheme for the optimal control of diffusion processes, *RAIRO, Modélisation Math. Anal. Numér.*, 29:97–122.
- Candler, G.V., 2001. 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.
- Capuzzo Dolcetta, I., 1983. On a discrete approximation of the Hamilton-Jacobi equation of dynamic programming. *Appl. Math. Optim.*, 10:367–377.
- Capuzzo Dolcetta, I. and M. Falcone, 1989. Discrete dynamic programming and viscosity solutions of the Bellman equation. *Ann. Inst. Henri Poincaré, Anal. Non Linéaire*, 6 (supplement):161–184.
- Carlini, E., M. Falcone, and R. Ferretti, 2002. An efficient algorithm for Hamilton–Jacobi equations in high dimensions. Report 2002/7, Università di Roma “La Sapienza”, Dipartimento di Matematica “Guido Castelnuovo”. Submitted.
- Chow, C.-S. and J.N. Tsitsiklis, 1991. An optimal one–way multigrid algorithm for discrete–time stochastic control. *IEEE Trans. Autom. Control* 36:898–914.
- Daniel, J.W., 1976. Splines and efficiency in dynamic programming, *J. Math. Anal. Appl.* 54:402–407.

- Deissenberg, D., G. Feichtinger, W. Semmler, and F. Wirl, 2001. History dependence and global dynamics in models with multiple equilibria. Working Paper No. 12, Center for Empirical Macroeconomics, University of Bielefeld, [www.wiwi.uni-bielefeld.de](http://www.wiwi.uni-bielefeld.de).
- Falcone, M., 1987. A numerical approach to the infinite horizon problem of deterministic control theory. *Appl. Math. Optim.*, 15:1–13. *Corrigenda*, *ibid.*, 1991. 23: 213–214.
- Falcone, M. and R. Ferretti, 1998. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM J. Numer. Anal.* 35:909–940.
- Falcone, M. and T. Giorgi, 1999. An approximation scheme for evolutive Hamilton-Jacobi equations. In W. M. McEneaney, G.G. Yin, and Q. Zhang, editors, *Stochastic analysis, control, optimization and applications*, pages 288–303. Birkhäuser, Boston.
- Feichtinger, G., J.L. Haunschmied, P.M. Kort, and R.F. Hartl, 2000. A DNS-curve in a two state capital accumulation model: a numerical analysis. Forschungsbericht 242 des Instituts für Ökonometrie, OR und Systemtheorie, University of Technology, Vienna. Submitted.
- Feichtinger, G., P. Kort, R. F. Hartl, and F. Wirl, 2001. The dynamics of a simple relative adjustment–cost framework. *German Economic Review*, 2(3):255–268.
- González, R.L.V. and C. A. Sagastizábal, 1990. Un algorithme pour la résolution rapide d'équations discrètes de Hamilton-Jacobi-Bellman. *C. R. Acad. Sci., Paris, Sér. I*, 311:45–50.
- Grüne, L., 1997. An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation. *Numer. Math.*, 75(3):319–337.
- Grüne, L., 2003. Error estimation and adaptive discretization for the discrete stochastic Hamilton-Jacobi-Bellman equation. Preprint, Universität Bayreuth, submitted.
- Grüne, L., M. Kato, and W. Semmler, 2002. Numerical study of an ecological management problem. CEM Bielefeld, working paper.
- Grüne, L., M. Metscher, and M. Ohlberger, 1999. On numerical algorithm and interactive visualization for optimal control problems. *Computing and Visualization in Science*, 1(4):221–229.
- Grüne, L., W. Semmler, and M. Sieveking, 2001. Thresholds in a credit market model with multiple equilibria. Discussion Paper Nr. 482, Department of Economics, University of Bielefeld. Submitted.
- Hartl, R.F., P. Kort, G. Feichtinger and F. Wirl, 2000. Multiple equilibria and thresholds due to relative investment costs: non–concave–concave, focus–node, continuous–discontinuous, University of Technology, Vienna, working paper.
- Jermann, U.J., 1998. Asset pricing in production economies, *Journal of Monetary Economics* 41:257–275.
- Johnson, S.A., J.R. Stedinger, C.A. Shoemaker, Y. Li, J.A. Tejada–Guibert, 1993. Numerical solution of continuous–state dynamic programs using linear and spline interpolation, *Oper. Research* 41:484–500.
- Judd, K.L., 1996, 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.

- Judd, K.L., 1998. *Numerical methods in economics*. MIT Press, Cambridge.
- Judd, K.L. and S.-M. Guu, 1997. Asymptotic methods for aggregate growth models. *Journal of Economic Dynamics & Control* 21: 1025-1042.
- Keane, M.P. and K.I. Wolpin, 1994. 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.
- Ljungqvist, L. and T.J. Sargent, 2001. Recursive macroeconomic theory. MIT Press Cambridge, Mass..
- 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.
- Munos, R. and A. Moore, 2002. Variable resolution discretization in optimal control, *Machine Learning* 49:291–323.
- Reiter, M., 1999. Solving higher–dimensional continuous–time stochastic control problems by value function regression, *J. Econ. Dyn. Control*, 23:1329–1353.
- Santos, M.S. and J. Vigo–Aguilar 1995. Accuracy estimates for a numerical approach to stochastic growth models. Discussion Paper 107, Institute for Empirical Macroeconomics, Federal Reserve Bank of Minneapolis.
- Rust, J., 1996. Numerical Dynamic Programming in economics, in: Amman, H.M., D.A. Kendrick and J. Rust, eds., *Handbook of Computational Economics*, Elsevier, pp. 620–729.
- Rust, J., 1997. Using randomization to break the curse of dimensionality, *Econometrica* 65:478–516.
- Santos, M.S. and J. Vigo–Aguilar, 1998. Analysis of a numerical dynamic programming algorithm applied to economic models. *Econometrica*, 66(2):409–426.
- Seeck, A., 1997. Iterative Lösungen der Hamilton-Jacobi-Bellman-Gleichung bei unendlichem Zeithorizont. Diplomarbeit, Universität Kiel.
- Sieveking, M. and W. Semmler, 1997. The Present value of resources with large discount rates. *Applied Mathematics and Optimization*, 35: 283-309.
- Semmler, W. and M. Sieveking, 1994. On optimal exploitation of interacting resources. *Journal of Economics*, 59(1):23–49.
- Skiba, A.K., 1978. Optimal growth with a convex-concave a production function. *Econometrica*, 46, no. 3: 527-539.
- Tauchen, G. and R. Hussey, 1991. Quadrature–based methods for obtaining approximate solutions to nonlinear asset–price models, *Econometrica*, 59:371–396.
- Taylor J.B. and H. Uhlig, 1990. Solving non-linear stochastic growth models: a comparison of alternative solution methods. *Journal of Business and Economic Studies*, 8: 1-18.
- Trick, M.A. and S.E. Zin, 1993. A linear programming approach to solving stochastic dynamic programs, Working Paper, Carnegie–Mellon University.
- Trick, M.A. and S.E. Zin, 1997. Spline approximations to value functions: a linear programming approach, *Macroeconomic Dynamics*, 1:255-277.