Chapter 1

Adaptive grid generation for evolutive Hamilton-Jacobi-Bellman equations

Lars Grüne, Fachbereich Mathematik, J.W. Goethe–Universität, Postfach 111932, 60054 Frankfurt, Germany, gruene@math.uni-frankfurt.de

Abstract: We present an adaptive grid generation for a class of evolutive Hamilton-Jacobi-Bellman equations. Using a two step (semi–Lagrangian) discretization of the underlying optimal control problem we define a–posteriori local error estimates for the discretization error in space. Based on these estimates we present an iterative procedure for the generation of adaptive grids and discuss implementational details for a suitable hierarchical data structure.

Keywords: adaptive space discretization, Hamilton-Jacobi-Bellman equation, semi-Lagrangian scheme

1.1 Introduction

In the numerical approximation of partial differential equations one of the main sources of computational cost is the number of nodes used for the space discretization, which determines the dimension of the finite dimensional problem to be solved. Two basic strategies are used in order to minimize the number of nodes needed for an accurate computation: high–order interpolation techniques in space and adaptive gridding techniques (and, of course, combinations of both).

In this paper we focus on the adaptive gridding technique for the fol-

lowing nonlinear first order Hamilton-Jacobi-Bellman equation

$$\begin{aligned} &\frac{\partial}{\partial t}v(x,t) + \lambda v(x,t) + \sup_{u\in U}\{-f(x,u)\cdot\nabla v(x,t) - g(x,u)\} = 0 \\ &v(x,0) = v^0(x) \end{aligned} \tag{1.1}$$

for $(x,t) \in \Omega \times [0,T]$ where Ω is an open and bounded subset of $\mathbb{R}^n$.

This equation is related to a finite horizon optimal control problem (see [1] for an extensive discussion of this relation), and it is well known that in general one has to consider viscosity solutions of (1.1) as smooth solutions do not exist in general. The scheme we are going to use has its origins in the numerical approximation of the infinite horizon problem. Based on a discrete time approximation of the underlying optimal control problem [2; 3] a subsequent space discretization [4; 9] yields the fully discrete scheme. This scheme is easily adapted to the evolutive equation and for equidistant space discretization a convergence analysis was carried out in [7].

A particular subclass of (1.1) are the linear advection equations which are obtained if the functions f and g do not depend on u. In this case, the resulting scheme is also known as "semi–Lagrangian approximation" which was introduced in [12] and has been extensively used in the simulation of models for weather forecast and oceanography, see e.g. the review [13]. Although this scheme has been applied for quite a while, a rigorous error analysis appeared only recently in [6]. In this reference high–order approximations are investigated, and it is remarked (see [6, Example 5]) that for nonsmooth solutions these high–order approximations in space in general only yield first order convergence rates. It is primarily this situation where we expect an adaptive approach based on a first–order approximation to be advantageous and the numerical tests in the last section confirm this expectation.

This paper is organized as follows. First, we introduce the time and space discretization in Section 1.2, and prove a useful regularity result for the fully discrete approximation. After that, in Section 1.3 we turn to the definition of local error estimates and prove a number of properties of these values. The basic idea in this part stems from [9] where the analogous results are given for the stationary equation, which corresponds to the infinite horizon optimal control problem. In Section 1.4 we turn to the implementation of the scheme exploiting in particular properties of a suitable hierarchical data structure. Finally, we end this paper with two numerical examples in Section 1.5.

1.2 Discretization in time and space

In this section we describe the discretization for equation (1.1). Let us first state the assumption on the data of the problem. We assume that $U \subset \mathbb{R}^m$ is compact, v^0 is Lipschitz with constant L_0 and the functions $f : \Omega \times U \to \mathbb{R}^n$ and $g : \Omega \times U \to \mathbb{R}$ are norm–bounded by constants M_f and M_g for all $u \in U$, continuous in both variables and Lipschitz in x with constants L_f and L_g uniformly in $u \in U$.

Let $\varphi(t, x, u)$ denote the (unique) solution of the initial value problem

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x,$$

where $u \in \mathcal{U} := \{u : \mathbb{R} \to U \mid u \text{ measurable}\}$. For simplicity of presentation we assume that Ω is forward invariant under f, i.e. $\varphi(t, x, u) \in \Omega$ for all $x \in \Omega$, all $t \geq 0$ and all $u \in \mathcal{U}$. Note that this assumption can be relaxed by imposing a suitable boundary condition on the inflowing part of Ω, see [6, Section 1]. Furthermore, our setup is easily extended to min–max problems.

A standard dynamic programming argument (see e.g. [1]) shows that for each $\tau > 0$ and each $t \in [0, T - \tau]$ the solution v is uniquely determined by the dynamic programming principle

$$v(x, t + \tau) = \inf_{u \in \mathcal{U}} \left\{ \int_0^\tau e^{\lambda s} g(\varphi(s, x, u), u(s)) ds + e^{\lambda \tau} v(\varphi(\tau, x, u), t) \right\} \quad (1.2)$$

The discretization of (1.1) now relies on a discretization of (1.2). For positive time steps $h > 0$ we pick a family of finite sets of control values U_h, and a numerical one step scheme Φ_h (satisfying the usual consistency and stability conditions) as well as a numerical integration scheme I_h such that for each $u \in \mathcal{U}$ there exists $u_h \in U_h$, and for each $u_h \in U_h$ there exists $u \in \mathcal{U}$ with

$$\|\varphi(h, x, u) - \Phi_h(x, u_h)\| \leq C h^{p+1}$$

and

$$\left| \int_0^h e^{\lambda s} g(\varphi(s, x, u), u(s)) ds - I_h(x, u) \right| \leq C h^{p+1}$$

for some $C > 0$ which is independent of h.

In the case where f and g do not depend on u this can be done e.g. by a Runge–Kutta scheme and an associated quadrature rule, cfr. [6], for explicitly u–dependent data several schemes are presented e.g. in [5], [8] or

[10] (note that although [5] deals with the stationary equation the results are easily adapted to our setup here).

Again, for simplicity, we assume forward invariance of Ω with respect to the discretized dynamics, i.e. $\Phi_h(x,u) \in \Omega$ for all $u \in U_h$, all $x \in \Omega$ and all $h > 0$.

Using these schemes we can replace (1.2) by the discrete time approximation

$$v_h(x, t_{i+1}) = T_h(v_h)(x, t_i) \tag{1.3}$$

with $t_i = ih$ and $i \geq 0$, where for any function $w : \Omega \times h\mathbb{N}_0 \to \mathbb{R}$ the operator T_h is given by

$$T_h(w)(x,t) := \inf_{u \in U_h} \left\{ I_h(x,u) + e^{\lambda h} w(\Phi_h(x,u), t) \right\} \tag{1.4}$$

We will also refer to (1.3) as the discrete evolutive Hamilton–Jacobi–Bellman equation.

Standard arguments (see [5] or [6]) imply the existence of a constant $K > 0$ such that

$$\sup_{x \in \Omega} |v_h(x, t_i) - v(x, t_i)| \leq K h^p \text{ for all } t_i \in [0, T].$$

Our main concern in this paper is the discretization of (1.3) in space. Instead of an equidistant high–order space discretization as considered e.g. in [6] here we are going to use a first order approximation along with a suitable locally refined grid. In the remainder of this section we introduce the class of approximations we are going to use and prove a useful regularity result for these approximations.

Since for the problems under consideration the domain Ω usually has a rather simple shape (or the problem can be transformed to some simply shaped Ω), for the discretization in space we use a partition of Ω into cubic elements. This structure has several implementational advantages (e.g. for a given point x the surrounding element is easily determined by integer computations, the gridding procedure works for arbitrary dimensions without extra effort, no linear equations have to be solved for the interpolation), and a certain built in regularity which is exploited in Lemma 1.1, below. Nevertheless, most of the following considerations also apply to triangular grids, see e.g. [9] where Lemma 1.1 is proved for triangular grids under suitable regularity conditions.

Formally, the cubic elements are of the form

$$Q_j := \{x = (x^1, \ldots, x^n)^T \in \mathbb{R}^n \,|\, x^i \in [a_j^i, b_j^i] \text{ for all } i = 1, \ldots, n\} \quad (1.5)$$

for real values $a_j^i < b_j^i$ for all $i = 1, \ldots, n$. The nodes of an element are given by the set

$$\text{nodes}(Q_j) := \{x \in \mathbb{R}^n \,|\, x^i = a_j^i \text{ or } x^i = b_j^i \text{ for all } i = 1, \ldots, n\}$$

A grid Γ is a collection of P cubic elements Q_j such that $\text{cl}\Omega = \bigcup_{j=1,\ldots,P} Q_j$ and $\text{int}Q_j \cap \text{int}Q_k = \emptyset$ for all $j \neq k$. The nodes x_l of Γ are given by

$$\text{nodes}(\Gamma) := \bigcup_{j=1,\ldots,P} \text{nodes}(Q_j),$$

and N denotes the number of nodes.

A node x_l is called *regular* if $x_l \in \text{nodes}(Q_j)$ for all j with $x_l \in Q_j$, otherwise it is called *hanging*. We denote the set of regular nodes of Γ by $\text{reg}(\Gamma)$. Figure 1.1 show a 2d grid in which the hanging nodes are marked.

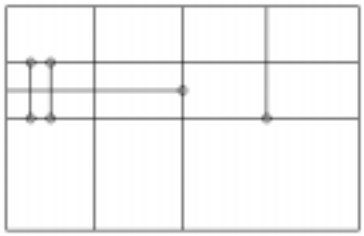

Fig. 1.1 Hanging nodes in a 2d grid

On a grid Γ we consider the space of multilinear functions

$$W := \{w \in C(\text{cl}\Omega) \,|\, w(x + \alpha e_k) \text{ is linear in } \alpha \text{ on each } Q_j \text{ for each } k\}$$

where the e_k, $k = 1, \ldots, n$ denote the standard basis vectors of $\mathbb{R}^n$.

Note that the the assumption $w \in C(\text{cl}\Omega)$ (i.e. w continuous) implies that w is uniquely determined by its values in the regular nodes $\text{reg}(\Gamma)$, and the values in the hanging nodes have to be determined by linear interpolation. For some element Q_j of Γ with nodes $x_l = (x_l^1, \ldots, x_l^n)^T$,

$l = 1, \ldots, 2^n$ and coordinates $a_j^i < b_j^i$, $i = 1, \ldots, n$, the value $w(x)$ for $x = (x^1, \ldots, x^n)^T \in Q_j$ is given by the multilinear interpolation

$$w(x) = \sum_{l=1}^{2^n} \mu_l(x) w(x_l)$$

where

$$\mu_l(x) = \prod_{i=1}^{n} g_{li}(x^i) \quad \text{with} \quad g_{li}(x^i) = \begin{cases} (x^i - a_j^i)/(b_j^i - a_j^i), & \text{if } x_l^i = b_j^i \\ (b_j^i - x^i)/(b_j^i - a_j^i), & \text{if } x_l^i = a_j^i \end{cases}$$

Remark 1.1 *For dimensions $n \geq 3$ a grid needs to satisfy some regularity conditions in order to ensure the existence of a continuous function $w \in W$ for arbitrary given values in the regular nodes. Consider, for example, a three dimensional grid with two touching elements as in Figure 1.2.*

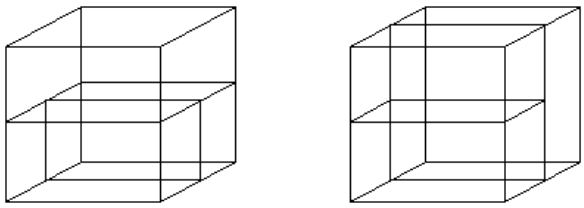

Fig. 1.2 Two touching elements in a 3d grid

In the situation of Figure 1.2 the value in the central node of the touching faces could either be determinated by interpolating the values in its vertical neighbors (which one would do by looking at the right cube) or by interpolating the values in its horizontal neighbors (as one would do in the left cube). Since these values in general do not coincide, one needs some condition to avoid these situations.

A sufficient regularity condition for this purpose is for instance given by

$$(\text{nodes}(Q_j) \cup \text{nodes}(Q_k)) \cap (Q_j \cap Q_k) \subset \text{nodes}(Q_l) \text{ for } l = j \text{ or } l = k$$

for all $j, k = 1, \ldots, P$ with $j \neq k$, which means that for each two touching elements there is a unique finer one.

Furthermore, in order to reduce the number of hanging nodes it is useful to avoid large differences in size of each two neighboring elements.

Using the space W we can now define a finite dimensional approximation to v_h. For a sequence of grids Γ^i, $i \geq 0$ we set

$$v_h^{\Gamma^0}(x_l, 0) = v^0(x_l), \quad v_h^{\Gamma^i}(x_l, t_i) = T_h(v_h^{\Gamma^{i-1}})(x_l, t_{i-1}) \tag{1.6}$$

for $i \geq 1$, $t_i = ih \leq T$, and all regular nodes $x_l \in \mathrm{reg}(\Gamma^i)$. Note that the evaluation of T_h needs the values of $v_h^{\Gamma^{i-1}}$ at the points $\Phi_h(x_l, u)$, which in general will not be nodes of the grid; hence these values have to be determined by interpolation.

The discretization error of this approximation will be investigated in the next section. Before turning to this analysis, we will investigate the continuity properties of $v_h^{\Gamma^i}$ in x. For this we need the following Lemma, whose proof is straightforward using the definition of W.

Lemma 1.1 *Let $w \in W$ be an arbitrary function on some grid Γ. Then the estimate*

$$\frac{|w(x) - w(y)|}{\|x - y\|_1} \leq \max_{\substack{\tilde{x}, \tilde{y} \in \mathrm{nodes}(\Gamma) \\ \tilde{x} \neq \tilde{y}}} \frac{|w(\tilde{x}) - w(\tilde{y})|}{\|\tilde{x} - \tilde{y}\|_1}$$

for all $x, y \in \Omega$ with $x \neq y$. Here $\|\cdot\|_1$ denotes the 1–norm given by $\|x\|_1 := \sum_{i=1}^n |x^i|$.

Based on this Lemma we can now derive a Lipschitz estimate for the approximations $v_h^{\Gamma^i}$. For this we have to make the following assumption on the numerical schemes I_h and Φ_h: There exist constants L_{I_h}, $L_{\Phi_h} > 0$ such that

$$|I_h(x,u) - I_h(y,u)| \leq L_{I_h}\|x-y\|_1 \text{ and } |\Phi_h(x,u) - \Phi_h(y,u)| \leq L_{\Phi_h}\|x-y\|_1 \tag{1.7}$$

for all $x, y \in \Omega$ and all $u \in U_h$. Note that if f and g in (1.1) are Lipschitz in $x \in \Omega$ uniformly in $u \in U$ then most reasonable numerical schemes (like quadrature rules for I_h and Runge–Kutta schemes for Φ_h) satisfy (1.7) for suitable constants. Recall that we also assumed v^0 to be Lipschitz with constant L_0, without loss of generality we may assume this Lipschitz estimate to be valid in the 1–norm.

Proposition 1.1 *Assume (1.7), let Γ^i be a sequence of grids and consider the functions $v_h^{\Gamma^i}$ defined by (1.6). Then the estimate*

$$|v_h^{\Gamma^i}(x, t_i) - v_h^{\Gamma^i}(y, t_i)| \leq L_i\|x - y\|_1$$

holds, where L_i is defined inductively by $L_{i+1} = L_{I_h} + e^{\lambda h} L_{\Phi_h} L_i$, and L_0 is the Lipschitz constant of v^0 with respect to the 1–norm.

Proof. Let $w : \Omega \times h\mathbb{N}_0 \to \mathbb{R}$ be a function which for some $t = ih$, $i \in \mathbb{N}_0$, is Lipschitz in the first argument with constant L. Then the operator T_h satisfies

$$\begin{aligned}
&|T_h(w)(x,t) - T_h(w)(y,t)| \\
&= \Big| \inf_{u\in U_h} \big\{ I_h(x,u) + e^{\lambda h} w(\Phi_h(x,u),t) \big\} \\
&\quad - \inf_{u\in U_h} \big\{ I_h(y,u) + e^{\lambda h} w(\Phi_h(y,u),t) \big\} \Big| \\
&\le \sup_{u\in U_h} \big| I_h(x,u) + e^{\lambda h} w(\Phi_h(x,u),t) - I_h(y,u) - e^{\lambda h} w(\Phi_h(y,u),t) \big| \\
&\le \sup_{u\in U_h} |I_h(x,u) - I_h(y,u)| + e^{\lambda h} |w(\Phi_h(x,u),t) - w(\Phi_h(y,u),t)| \\
&\le L_{I_h} \|x-y\|_1 + e^{\lambda h} L_{\Phi_h} L \|x-y\|_1 \;=\; (L_{I_h} + e^{\lambda h} L_{\Phi_h} L) \|x-y\|_1
\end{aligned}$$

for all points $x, y \in \Omega$. Hence the assertion follows by induction using Lemma 1.1. □

1.3 Error estimation

In this section we present an a–posteriori error estimator for the space discretization (1.6) of the discrete evolutive Hamilton–Jacobi equation (1.3).

We define the following errors for the approximations $v_h^{\Gamma^i}(\cdot, t_i)$ on a sequence of grids Γ^i.

Definition 1.1 We define the local error for each time $t_i = ih$, $i \ge 1$ by

$$\eta_{loc}^i := \sup_{x\in\Omega} |v_h^{\Gamma^i}(x,t_i) - T_h(v_h^{\Gamma^{i-1}})(x,t_{i-1})|$$

and for $i = 0$ by

$$\eta_{loc}^0 := \sup_{x\in\Omega} |v_h^{\Gamma^0}(x,t_0) - v^0(x)|$$

The global error for each time $t_i = ih$, $i \ge 0$ is defined by

$$\eta_{glob}^i := \sup_{x\in\Omega} |v_h^{\Gamma^i}(x,t_i) - v_h(x,t_i)|.$$

Note that while these expressions measure the error in space, the terms "local" and "global" here refer to time, i.e., η^i_{loc} measures the spatial error introduced in one time–step, while η^i_{glob} measures the accumulated error over the whole time interval. The following definition refines this concept to local error estimates in space.

Definition 1.2 For each $x \in \Omega$ we define the pointwise local error by

$$\eta^0(x) := \left| v_h^{\Gamma^0}(x,0) - v^0(x) \right|, \quad \eta^i(x) := \left| v_h^{\Gamma^i}(x,t_i) - T_h(v_h^{\Gamma^{i-1}})(x,t_{i-1}) \right|$$

and for each element Q_j of Γ^i we define the elementwise local error by

$$\eta^i_j := \sup_{x \in Q_j} \eta^i(x).$$

The following Theorem shows the relation of this value to the errors from Definitions 1.1 and 1.2.

Theorem 1.1 *Let Γ^i be a sequence of grids with P^i elements and let $v_h^{\Gamma^i}(\cdot,t_i)$ denote the corresponding solutions. Then the following inequalities hold*

$$\begin{aligned}
\eta^i_{loc} &= \sup_{j=1,\dots,P^i} \eta^i_j \\
\eta^i_{glob} &\le \sum_{k=0}^{i} e^{\lambda h(i-k)} \eta^k_{loc} \\
&= \sum_{k=0}^{i} e^{\lambda h(i-k)} \sup_{j=1,\dots,P^k} \eta^k_j \\
&\le C(\lambda,h,i) \sup_{k=0,\dots,i,\ j=1,\dots,P^k} \eta^k_j,
\end{aligned}$$

where $C(\lambda,h,i) = i+1$ if $\lambda = 0$ and $C(\lambda,h,i) = (1-e^{\lambda h(i+1)})/(1-e^{\lambda h})$ if $\lambda \neq 0$.

Proof. Similarly to the proof of Proposition 1.1 one sees that for each two continuous functions $w_1,\ w_2 : \Omega \times h\mathbb{N}_0 \to \mathbb{R}$ the operator T_h satisfies

$$|T_h(w_1)(x,t) - T_h(w_2)(x,t)| \le e^{\lambda h} \sup_{x \in \Omega} |w_1(x,t) - w_2(x,t)|$$

for all $x \in \Omega$ and all $t > 0$. Hence the assertion follows immediately by induction from the definition of the errors and the identity $v_h(x,t_i) = T_h(v_h)(x,t_{i-1})$. □

A natural guess for an adaptive strategy is now the following procedure:

For $i \geq 0$, given some grid Γ^i with P^i elements, and some tolerance $tol > 0$ for the local error, compute $v_h^{\Gamma^i}(\cdot,t_i)$ (based on $v_h^{\Gamma^{i-1}}(\cdot,t_{i-1})$ or v^0) and the errors η_j^i, $j = 1,\ldots,P^i$, refine all elements Q_j with $\eta_j^i > tol$. Repeat this procedure iteratively until $\eta_j^i \leq tol$ for all $j = 1,\ldots,P^i$. (A more detailed description of this procedure which also incorporates coarsening of those elements with small errors is given below in Section 1.4.)

In order to ensure that the local error estimates η_j^i eventually become small (i.e. to ensure termination of this iteration), we need a relation between the size of the element Q_j and the size of the local error η_j^i. The following lemma gives the necessary estimate

Lemma 1.2 *Assume* (1.7). *Then*

$$|\eta^i(x) - \eta^i(y)| \leq L_i \|x-y\|_1$$

for all $x, y \in \Omega$ *with* L_i *from Proposition 1.1.*

Proof. By the definition of the error estimator we have that

$$\begin{aligned}&|\eta^i(x) - \eta^i(y)|\\ &\quad\leq |v_h^{\Gamma^i}(x,t_i) - v_h^{\Gamma^i}(y,t_i)| + |T_h(v_h^{\Gamma^{i-1}})(x,t_{i-1}) - T_h(v_h^{\Gamma^{i-1}})(x,t_{i-1})|.\end{aligned}$$

Since the second term can be estimated as in the proof of Proposition 1.1, the assertion follows from the Lipschitz estimates of $v_h^{\Gamma^i}(\cdot,t_i)$ and $v_h^{\Gamma^{i-1}}(\cdot,t_{i-1})$ provided by Proposition 1.1. □

The relation between the size of Q_j and η_j^i is now given by the following proposition.

Proposition 1.2 *Assume* (1.7). *Then*

$$\eta_j^i \leq L_i \sup_{x,y \in Q_j} \|x-y\|_1$$

with L_i *from Proposition 1.1.*

Proof. Since $\eta^i(x) = 0$ for all nodes of the element Q_j we immediately obtain the desired inequality by Lemma 1.2. □

Remark 1.2 *In particular, these estimates show the convergence of the scheme as the size of the elements tends to 0. More precisely, defining* $\Delta x := \max_{j=1,\dots,P} \operatorname{diam}(Q_j)$ *(where the diameter diam is measured in the* 1*–norm), and assuming the grid to be constant, we obtain*

$$\eta^i_{glob} \le C(\lambda, h, i) \max_{j=0,\dots,i} L_j \Delta x$$

for the constant $C(\lambda, h, i)$ *from Theorem 1.1. Note, however, that for the full discretization error (i.e., the error between* v^Γ_h *and* v*) refined estimates based on viscosity techniques have been obtained in [7].*

When looking at the definition of η^i_j one sees that the computation of this value involves the evaluation of $T_h(v_h^{\Gamma^{i-1}})$ at infinitely many points. Of course, this is no feasible task, however, again exploiting Lemma 1.2 we can see that η^i_j can be approximated by some value $\bar\eta^i_j$ which is computable. For this task consider an element Q_k and a collection of test points $\bar x_l \in Q_j$, $l = 1, \dots, p$. Defining

$$\bar\eta^i_j := \max_{l=1,\dots,p} \eta(\bar x_l),$$

the following lemma shows that this value indeed gives a good approximation to η^i_j.

Lemma 1.3 *Assume* (1.7)*, and consider a finite set of points* $\bar x_l \in Q_j$*,* $l = 1, \dots, p$ *satisfying*

$$\sup_{x \in Q_j} \min_{l=1,\dots,p} \|x - x_l\|_1 \le \alpha$$

for some $\alpha > 0$*. Then*

$$|\bar\eta^i_j - \eta^i_j| \le L_i \alpha$$

for L_i *from Proposition 1.1.*

Proof. Using Lemma 1.2 we obtain

$$\begin{aligned} |\bar\eta^i_j - \eta^i_j| &\le \sup_{x \in Q_j} \min_{l=1,\dots,p} |\eta(x) - \eta(x_l)| \\ &\le L_i \sup_{x \in Q_j} \min_{l=1,\dots,p} \|x - x_l\|_1 \le L_i \alpha \end{aligned}$$

which shows the assertion. □

1.4 Implementation details

In this section we will describe several implementational issues of our adaptive scheme in greater detail. Our main focus will be the choice of the test points x_l for the numerical evaluation of the local error.

As already mentioned in the last section, the exact evaluation of the local errors η_j^i is in general not possible; instead by Lemma 1.3 it is justified to use an approximation $\bar{\eta}_j^i$ by an evaluation in finitely many test points $\bar{x}_l$ per cuboid.

In what follows we will indicate how a suitable choice of these test points can lead to an efficient implementation. Before going into details, we briefly recall some facts about hierarchical grids which form the basis for our particular choice.

The idea of a hierarchical grid is based on a successive local refinement of an equidistant (and coarse) grid. A locally refined grid is obtained by subdividing an element Q in one coordinate direction x^{i_0}, $i_0 \in \{1,\dots,n\}$. We denote this refinement by $\mathrm{ref}(Q,i_0)$. More precisely, if Q is given by (1.5) with coordinates $a^i < b^i$, $i = 1,\dots,n$ then $\mathrm{ref}(Q,i_0)$ consists of the two elements Q_1 and Q_2 with coordinates

$$\begin{aligned} a_1^i &= a^i \text{ for } i = 1,\dots,n,\\ b_1^i &= b^i \text{ for } i = 1,\dots,n \text{ with } i \neq i_0,\\ b_1^{i_0} &= a^{i_0} + (b^{i_0} - a^{i_0})/2 \end{aligned}$$

and

$$\begin{aligned} b_2^i &= a^i \text{ for } i = 1,\dots,n,\\ a_2^i &= a^i \text{ for } i = 1,\dots,n \text{ with } i \neq i_0,\\ a_2^{i_0} &= a^{i_0} + (b^{i_0} - a^{i_0})/2. \end{aligned}$$

For any element $\tilde{Q}$ of the refinement $\mathrm{ref}(Q,i_0)$ we call Q the coarsening of $\tilde{Q}$, formally $\mathrm{coarse}(\tilde{Q}) = Q$. For a refinement in all coordinate directions we write $\mathrm{ref}(Q)$ (i.e. for an n–dimensional grid $\mathrm{ref}(Q)$ consists of 2^n elements). The refinement of the whole grid is denoted by $\mathrm{ref}(\Gamma)$, and the m–th refinement by $\mathrm{ref}^m(\Gamma)$, i.e., $\mathrm{ref}^1(\Gamma) = \mathrm{ref}(\Gamma)$, $\mathrm{ref}^{m+1}(\Gamma) = \mathrm{ref}(\mathrm{ref}^m(\Gamma))$. Figure 1.3 shows the representation of this structure in a binary tree data structure, for more informations on this data structure and hierarchical grids we refer to [11].

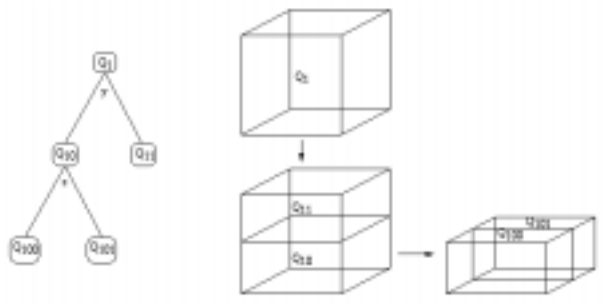

Fig. 1.3 Representation of a 3d refinement in a binary tree

The idea of an efficient choice of the test points is now, that along with some grid Γ we consider its m–th refinement $\mathrm{ref}(\Gamma, m)$ for some $m \geq 1$ and choose the test points $\mathrm{nodes}(\mathrm{ref}(\Gamma, m)) \setminus \mathrm{nodes}(\Gamma)$. Figure 1.4 shows the resulting points for $m = 1$ and $m = 2$ for a 2d element.

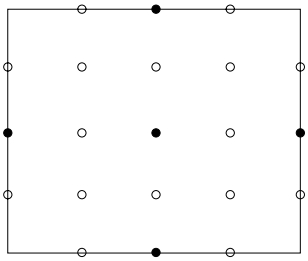

Fig. 1.4 Testpoints for $m = 1$ (black) and $m = 2$ (black and white) in 2d

Doing so, the values in the test points evaluated for the computation of $\bar{\eta}_j^i$ can be used in the next iteration and hence are not "lost" after $\bar{\eta}_j^i$ has been evaluated. More precisely, once a solution $v_h^{\Gamma^{i-1}}(\cdot, t_{i-1})$ is known the grid generation for the next iteration i is described in the following algorithm:

Step 1: Choose an initial grid Γ_0^i and parameters $m \geq 1$ and $tol > 0$; let $\tilde{\Gamma}_0^i := \mathrm{ref}(\Gamma_0^i, m)$ and $k = 0$

Step 2: Compute the solution $v_h^{\tilde{\Gamma}_k^i}(\cdot, t_i)$ on $\tilde{\Gamma}_k^i$

Step 3: Compute $\eta(x_l)$ for all $x_l \in \text{nodes}(\tilde{\Gamma}^i_k) \setminus \text{nodes}(\Gamma^i_k)$; if $\eta(x_l) \le tol$ for all x_l go to Step 5

Step 4: Choose a refinement Γ^i_{k+1} of Γ^i_k such that $x_l \in \text{nodes}(\Gamma^i_{k+1})$ for all x_l with $\eta(x_l) > tol$, and let $\tilde{\Gamma}^i_{k+1} := \text{ref}(\Gamma^i_{k+1}, m)$, $k = k+1$ and go to Step 2

Step 5: Coarsen all elements Q of Γ^i_k with $\bar{\eta}(\text{coarse}(Q)) \le tol$, regularize the grid if necessary (cf. Remark 1.1), and let Γ^i be the resulting grid and $\tilde{\Gamma}^i := \text{ref}(\Gamma^i, m)$.

The condition $\bar{\eta}(\text{coarse}(Q)) \le tol$ used for the coarsening in Step 5 ensures that all those nodes are removed from the grid which do not increase the accuracy by more than tol. In other words, all elements Q of the first refined and then coarsened grid satisfy $\bar{\eta}(Q) \le tol$ and thus fulfill the desired local accuracy bound.

The refinement iteration in the Steps 2–4 can be implemented very efficiently when the following facts are taken into account:

(i) Since the hierarchical data structure allows direct access to all subgrids of a given grid, all the operations can be done directly on $\tilde{\Gamma}^i_k$, i.e. Γ^i_k does not need to be stored seperately

(ii) After Step 4 is performed, the computation in Step 2 only involves the computation of $v_h^{\tilde{\Gamma}^i_k}$ in those nodes added in the preceding Step 4.

(iii) The evaluation of $\eta(x_l)$ can be done by comparing $v_h^{\tilde{\Gamma}^i_k}$ and $v_h^{\Gamma^i_k}$, which are both already computed. In particular, no additional evaluations of T_h (i.e. of f and g) are needed.

(iv) The function $v_h^{\tilde{\Gamma}^{i-1}}$ (instead of $v_h^{\Gamma^{i-1}}$) can be used in the computation in Step 2, giving additional accuracy.

(v) Γ^0 can be constructed the same way, where in Step 2 we use the initial value v^0 for the computation of $v_h^{\tilde{\Gamma}^0_k}$ (and omit the coarsening in Step 5).

Note that the choice of new nodes in this algorithm results in an anisotropic refinement, i.e. it is not required that an element is refined in all coordinate directions. For the choice of the initial grid Γ^i_0 there are several possibilities. In the numerical tests in the next section we have chosen the grid Γ^{i-1} from the last time step. A different possibility would be to start off from a given coarse initial grid in each iteration. In this case no

coarsening is necessary in Step 5.

1.5 Numerical examples

For our first numerical tests we consider the classical "rotating cone" benchmark problem, see e.g. [6]. The equation is given by

$$\frac{\partial}{\partial t}v(x,t) - f(x)\frac{\partial}{\partial x}v(x,t) = 0 \tag{1.8}$$

for $x = (x_1, x_2)^T \in \mathbb{R}^2$ with $f(x_1, x_2) = (-x_2, x_1)^T$.

The initial value was chosen as the "cut" paraboloid

$$v^0(x) = \max\left\{0.08 - \left\|x - \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}\right\|^2, 0\right\}.$$

Note that v^0 is Lipschitz, but not differentiable, and so is $v(\cdot, t)$ for each $t > 0$.

The system was discretized in time by a first order explicit Euler scheme with $h = 0.5$ and the initial grid Γ_0^0 was chosen to be equidistant with $\Delta x = 0.2$ (in the 1–norm), which results in $N = 441$ nodes. The parameter m determining the number of test nodes was set to 1.

	$tol = 0.1$		$tol = 0.01$		$tol = 0.001$		$tol = 0.0001$	
i	N^i	$\tilde{\eta}^i_{glob}$	N^i	$\tilde{\eta}^i_{glob}$	N^i	$\tilde{\eta}^i_{glob}$	N^i	$\tilde{\eta}^i_{glob}$
0	441	0.000000	537	0.000000	2187	0.000000	20889	0.000000
1	441	0.008798	514	0.008164	2246	0.001403	18919	0.000182
2	441	0.012077	479	0.012468	1928	0.001985	17338	0.000230
3	441	0.015598	477	0.013425	1921	0.001999	16725	0.000665
4	441	0.015028	475	0.014007	1844	0.002912	15644	0.000678
5	441	0.018611	463	0.014441	1802	0.004038	15076	0.000670
6	441	0.021677	449	0.013832	1785	0.003452	14410	0.000632
7	441	0.025200	451	0.016434	1768	0.003091	14059	0.000673
8	441	0.027362	459	0.016546	1614	0.005015	13561	0.000596
9	441	0.029064	459	0.021947	1628	0.005061	13880	0.001849
10	441	0.032706	459	0.021727	1698	0.005242	13583	0.001829

Table 1.1 Number of nodes N^i and global node error $\tilde{\eta}^i_{glob}$ for example (1.8)

Table 1.1 shows the resulting number of nodes N^i and the global error

$\tilde{\eta}^i_{glob}$ in the nodes of the grid Γ^i defined by

$$\tilde{\eta}^i_{glob} := \max_{x \in \mathrm{nodes}(\Gamma^i)} |v_h^{\Gamma^i}(x, t_i) - v_h(x, t_i)|$$

for the iterations $i = 0, 1, \ldots, 10$ (i.e. up to $t = 5$) and different values of *tol*. As expected for the first order scheme the number of nodes grows approximately like $1/tol$, which shows the efficiency of the estimate. Furthermore, for all tolerances the global error is of the order of the expected bound $10tol$, which indicates that the choice of the test points in this example gives satisfactory results. One can observe that the number of nodes tends to decrease during the iteration. This is mainly due to a "numerical diffusion" effect: During the iteration the steep regions tend to flatten (within the given accuracy bounds) and thus the number of nodes needed to ensure the given error tolerance becomes smaller.

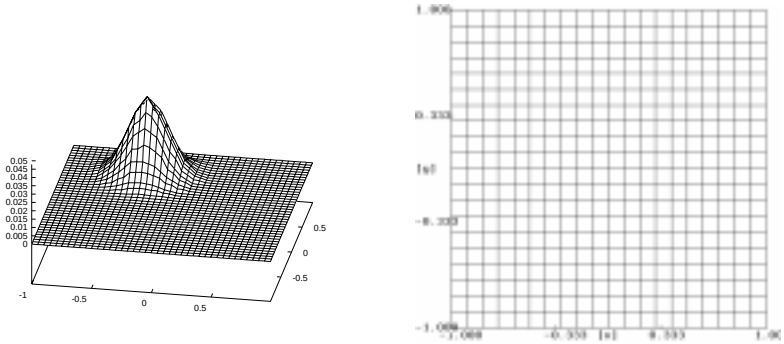

Fig. 1.5 Solution and grid after 10 iterations for example (1.8) with $tol = 0.1$

The Figures 1.5–1.8 show the solution $v_h^{\Gamma^{10}}(\cdot, 5)$ and the corresponding grid Γ^{10} after the last iteration for the different values of *tol*. In addition, Figure 1.9 shows the evolution of the grid during the computation for $tol = 0.001$. Observe that the effect of the usage of the preceding grid Γ^{i-1} in the construction of Γ^i is clearly visible.

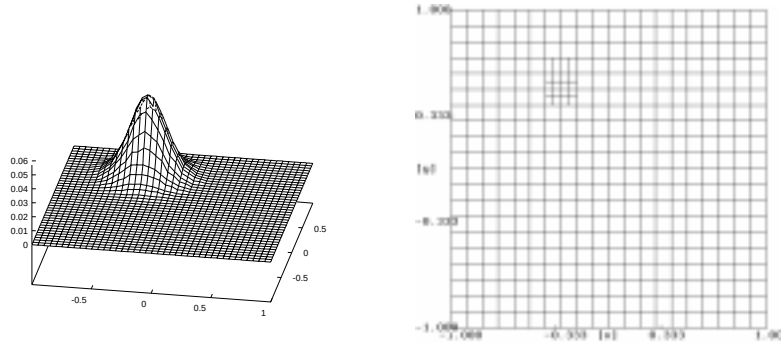

Fig. 1.6 Solution and grid after 10 iterations for example (1.8) with $tol = 0.01$

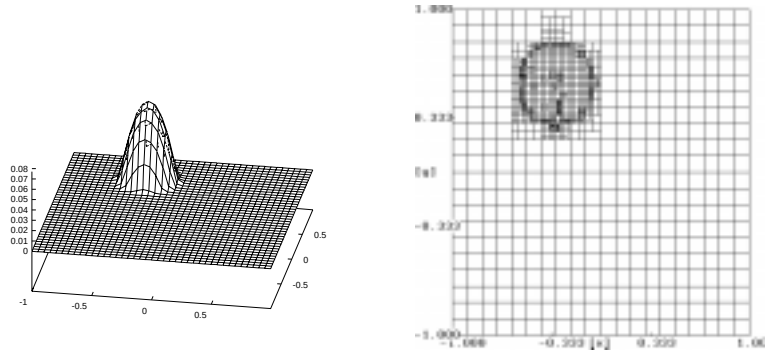

Fig. 1.7 Solution and grid after 10 iterations for example (1.8) with $tol = 0.001$

For our second test we consider a nonlinear variation of the first example. Here the equation is given by

$$\frac{\partial}{\partial t}v(x,t) + \inf_{u\in U}\left\{-f(x,u)\frac{\partial}{\partial x}v(x,t)\right\} = 0 \tag{1.9}$$

for $x = (x_1, x_2)^T \in \mathbb{R}^2$ with $f(x_1, x_2, u) = (-ux_2, ux_1)^T$ and one–dimensional control range $U = [0, 1]$. The initial value v^0 is chosen as before. Here the cone is not only rotating but also remains everywhere it has been before, i.e., is is "stretched" along the vectorfield.

For this test we used the same discretization in time as above, where the discrete set of control values U_h was chosen as $U_h = \{0, 1/9, \ldots, 8/9, 1\}$.

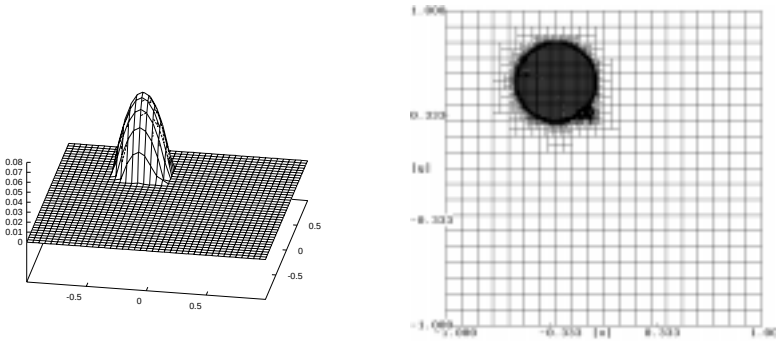

Fig. 1.8 Solution and grid after 10 iterations for example (1.8) with $tol = 0.0001$

Like for our first test, we show the results in Table 1.2, where we omit the error $\tilde{\eta}^i_{glob}$ for the smallest tolerance since the evaluation of the exact discrete time solution $v_h(x, t_i)$ for this nonlinear problem was not feasible for such a large number of nodes.

	$tol = 0.1$		$tol = 0.01$		$tol = 0.001$		$tol = 0.0001$
i	N^i	$\tilde{\eta}^i_{glob}$	N^i	$\tilde{\eta}^i_{glob}$	N^i	$\tilde{\eta}^i_{glob}$	N^i
0	441	0.000000	537	0.000000	2187	0.000000	25762
1	441	0.008798	578	0.008164	3627	0.001406	47293
2	441	0.012077	568	0.012468	4116	0.003712	55915
3	441	0.015598	575	0.012970	4524	0.004612	65767
4	441	0.014353	573	0.014990	4948	0.006488	75826
5	441	0.019241	549	0.020617	5283	0.007623	81893
6	441	0.022455	557	0.023129	5741	0.009144	90022
7	441	0.024617	543	0.025428	6279	0.010843	98775
8	441	0.026374	539	0.027224	6616	0.012378	105077
9	441	0.027804	539	0.028628	6944	0.013855	112625
10	441	0.029047	539	0.029883	7274	0.015125	122600

Table 1.2 Number of nodes N^i and global node error $\tilde{\eta}^i_{glob}$ for example (1.9)

The Figures 1.10–1.12 show the solutions after 10 iterations and the corresponding grid for $tol = 0.01$, 0.001 and 0.0001.

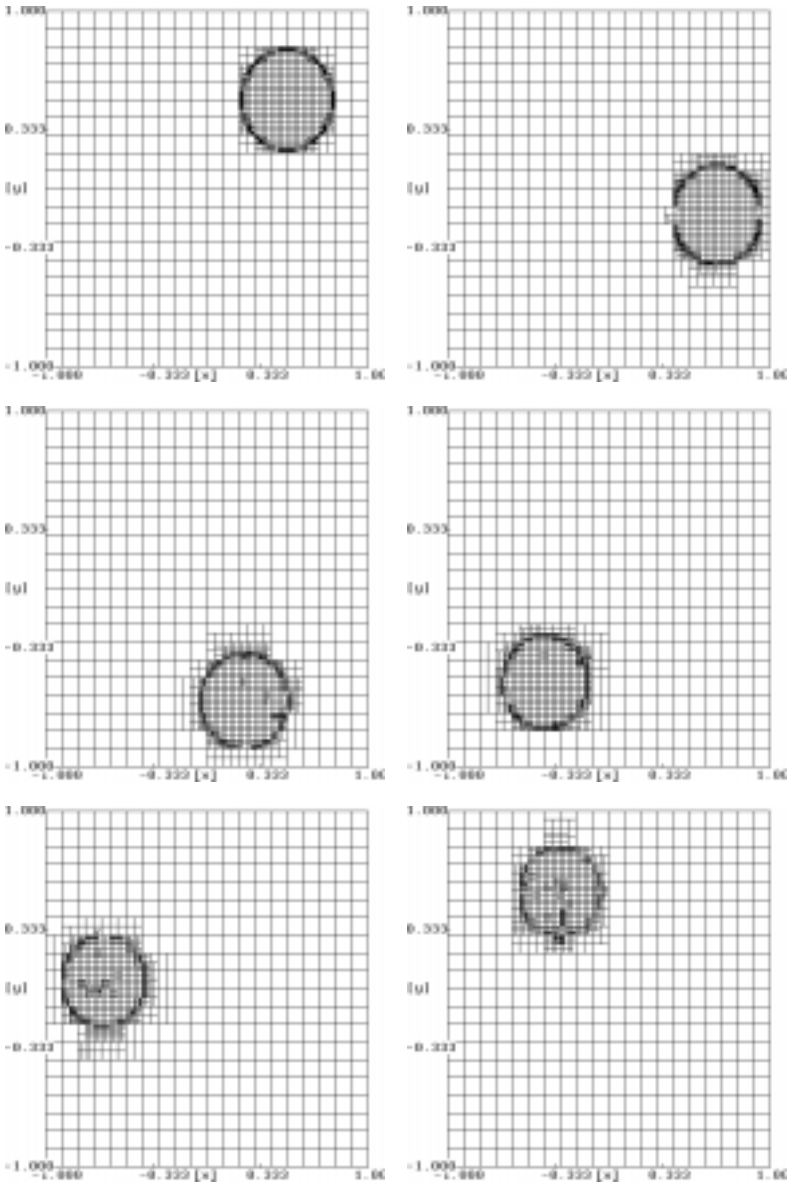

Fig. 1.9 Grid after 0, 2, 4, 6, 8, 10 iterations for example (1.8) with $tol = 0.001$

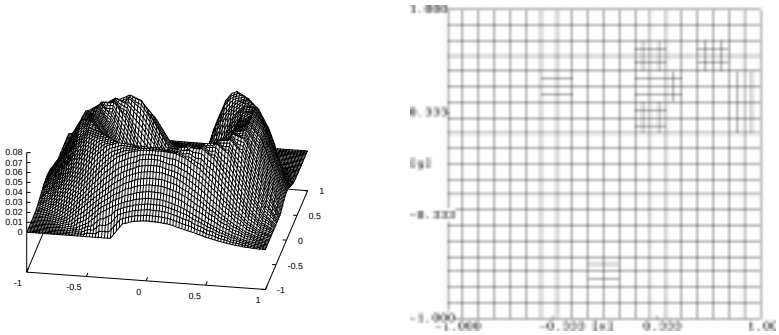

Fig. 1.10 Solution and grid after 10 iterations for example (1.9) with $tol = 0.01$

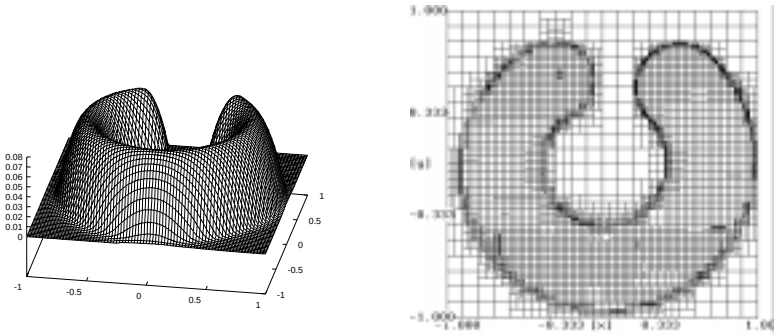

Fig. 1.11 Solution and grid after 10 iterations for example (1.9) with $tol = 0.001$

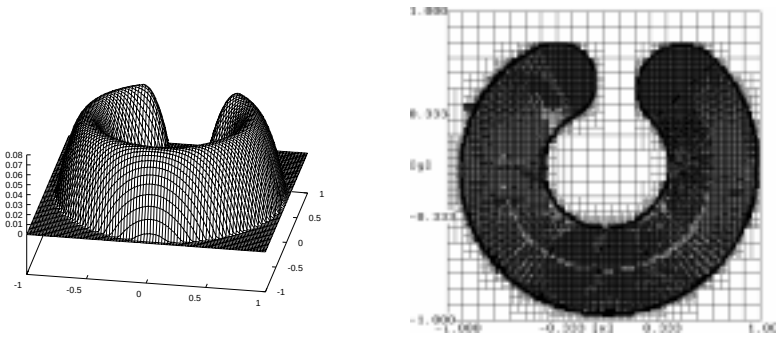

Fig. 1.12 Solution and grid after 10 iterations for example (1.9) with $tol = 0.0001$

Bibliography

1. M. Bardi and I. Capuzzo Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton–Jacobi–Bellman equations*, Birkhäuser, Boston, 1997.
2. I. Capuzzo Dolcetta, *On a discrete approximation of the Hamilton-Jacobi equation of dynamic programming*, Appl. Math. Optim., 10 (1983), pp. 367–377.
3. I. Capuzzo Dolcetta and H. Ishii, *Approximate solutions of the Bellman equation of deterministic control theory*, Appl. Math. Optim., 11 (1984), pp. 161–181.
4. M. Falcone, *A numerical approach to the infinite horizon problem of deterministic control theory*, Appl. Math. Optim., 15 (1987), pp. 1–13. *Corrigenda*, ibid. 23 (1991), 213–214.
5. M. Falcone and R. Ferretti, *Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations*, Numer. Math., 67 (1994), pp. 315–344.
6. M. Falcone and R. Ferretti, *Convergence analysis for a class of high-order semi-lagrangian advection schemes*, SIAM J. Numer. Anal., 35 (1998), pp. 909–940.
7. M. Falcone and T. Giorgi, *An approximation scheme for evolutive Hamilton-Jacobi equations*, in Stochastic analysis, control, optimization and applications, W. McEneaney et al., ed., Birkhäuser, Boston, 1999, pp. 288–303.
8. R. Ferretti, *High-order approximations of linear control systems via Runge-Kutta schemes*, Computing, 58 (1997), pp. 351–364.
9. L. Grüne, *An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation*, Numer. Math., 75 (1997), pp. 319–337.
10. L. Grüne and P. Kloeden, *Higher order numerical schemes for affinely controlled nonlinear systems*, Numer. Math., to appear.
11. L. Grüne, M. Metscher, and M. Ohlberger, *On numerical algorithm and interactive visualization for optimal control problems*, Computing and

Visualization in Science, 1 (1999), pp. 221–229.

12. A. Robert, *A stable numerical integration scheme for the primitive meteorological equations*, Atmos. Ocean., 19 (1981), pp. 35–46.
13. A. Staniforth and J. Coté, *Semi-Lagrangian integration schemes for atmospheric models—a review*, Monthly Weather Review, 119 (1991), pp. 2206–2223.