

Suboptimal Feedback Control of PDEs by Solving HJB Equations on Adaptive Sparse Grids

Jochen Garcke^{1,2} · Axel Kröner³

Received: 19 August 2015 / Revised: 26 April 2016 / Accepted: 14 June 2016 /
Published online: 24 June 2016
© Springer Science+Business Media New York 2016

Abstract An approach to solve finite time horizon suboptimal feedback control problems for partial differential equations is proposed by solving dynamic programming equations on adaptive sparse grids. A semi-discrete optimal control problem is introduced and the feedback control is derived from the corresponding value function. The value function can be characterized as the solution of an evolutionary Hamilton–Jacobi Bellman (HJB) equation which is defined over a state space whose dimension is equal to the dimension of the underlying semi-discrete system. Besides a low dimensional semi-discretization it is important to solve the HJB equation efficiently to address the curse of dimensionality. We propose to apply a semi-Lagrangian scheme using spatially adaptive sparse grids. Sparse grids allow the discretization of the value functions in (higher) space dimensions since the curse of dimensionality of full grid methods arises to a much smaller extent. For additional efficiency an adaptive grid refinement procedure is explored. The approach is illustrated for the wave equation and an extension to equations of Schrödinger type is indicated. We present several numerical examples studying the effect the parameters characterizing the sparse grid have on the accuracy of the value function and the optimal trajectory.

Keywords Closed-loop suboptimal control of PDEs · HJB equations · Sparse grids · Curse of dimensionality

Axel Kröner was supported by the Foundation Hadamard/Gaspard Monge Program for Optimization and Operations Research (PGMO) and the INRIA Explorer Programme.

✉ Axel Kröner
axel.kroener@inria.fr

Jochen Garcke
garcke@ins.uni-bonn.de

¹ Institut für Numerische Simulation, Universität Bonn, Bonn, Germany

² Fraunhofer SCAI, Sankt Augustin, Germany

³ INRIA Saclay and CMAP, École Polytechnique, Route de Saclay, 91128 Palaiseau Cedex, France

Mathematics Subject Classification 49K20 · 49N35 · 49L20 · 65D15

1 Introduction

In this paper we present a framework for finite horizon closed-loop suboptimal control of evolutionary partial differential equations (PDEs) based on a dynamic programming approach on adaptive sparse grids. We consider control problems for systems which arise from a semi-discretization (in space) of a PDE and solve the corresponding dynamic programming equations with adaptive semi-Lagrangian schemes on sparse grids. More precisely, we consider optimal control problems of the following type

$$\begin{cases} \min_{u \in U_{\text{ad}}} J(u) = \int_0^T l(y(t), u(t)) dt, \\ y_t(t) = f(y(t), u(t)), \\ y(0) = y_0, \end{cases} \quad (1.1)$$

with dynamics $f: \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}$, $d, m \in \mathbb{N}$, which arise from a semi-discretization of a PDE, running cost $l: \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}$, the initial state $y_0 \in \mathbb{R}^d$, the set of admissible controls U_{ad} , and time horizon $0 < T < \infty$. The presented approach is illustrated and numerically analyzed for the wave equation. Additionally we show how the approach can be transferred to equations of Schrödinger type and present numerical examples for a simplified bilinear setting in 2D. Although other approaches may be considered to control the solution of the corresponding semi-discrete systems—without using Hamilton–Jacobi Bellman (HJB) equations—the presented approach aims to give a general framework which is applicable on a wider class of problems. There are only few results on numerical methods for closed-loop optimal control of partial differential equations which are suitable for nonlinear problems.

To set up a feedback law we characterize the value function of problem (1.1) as the viscosity solution of an instationary HJB equation, from which we derive a control in feedback form. However, solving the HJB equation numerically is due to the curse of dimensionality very challenging in higher dimension. To reduce the computational effort there exist different possibilities. On the one hand the dimension of the dynamical system can be reduced by using model order reduction techniques for the discretization of the underlying PDE. An efficient reduction method is, e.g., proper orthogonal decomposition (POD) for certain classes of nonlinear equations, see, e.g., [32,33]. A low-dimensional discretization based on spectral elements (taking the first d sinus-modes as a basis of the discrete state space) is used in [31]. On the other hand efficient numerical methods are crucial for solving the HJB equation. There exists a wide range of methods including (higher order) finite difference [41], semi-Lagrangian [13,15], discontinuous Galerkin methods [28], sparse grids [10], or low rank tensor approximation for linear HJB equations [27].

In this paper we use spectral elements for the discretization of the underlying PDE following the approach in [31]. The corresponding HJB equation is solved by an adaptive semi-Lagrangian scheme on sparse grids based on [10]. While the HJB equation is defined on the full space, for the numerical approximation a finite computational domain and an artificial boundary condition has to be chosen carefully. The use of regular sparse grids implies a reduction of the degrees of freedom. For functions f from the Sobolev-space with dominating mixed derivative H_{mix}^2 the number of grid points on discretization level n , with one-dimensional mesh size $h_n = 2^{-n}$, reduces from $O(h_n^{-d})$ to $O(h_n^{-1}(\log h_n^{-1})^{d-1})$, whereas the asymptotic accuracy with respect to L^2 - or L^∞ -norm decreases only from $O(h_n^2)$

to $O(h_n^2(\log h_n^{-1})^{d-1})$. Sparse grids go back to the work by Smolyak [44] and in particular Zenger [47], Griebel [23], and Griebel and Bungartz [11]. In case of nonsmoothness of the solution an adaptive sparse grid scheme may allow to improve the accuracy of the approximation, we refer to [17, 43] for references on adaptive sparse grids. Note that the interpolation on sparse grids is not monotone [10, 43], e.g., the sparse grid interpolant of a non-negative function can be negative on non-interpolation points. Therefore the scheme is in general not monotone and a convergence theorem follows not directly from Barles and Souganidis [5].

In this paper we consider control problems for the wave and the bilinear Schrödinger equation. In contrast to [10] we consider different underlying systems and allow more general controls than bang-bang controls. We study numerically the error in the approximating value function (for $d = 2$) and the optimal trajectory and control ($2 \leq d \leq 8$). Since for normal sparse grids the nodes on the boundary become dominant in the computational complexity in higher dimensions, i.e. the ratio of points on the boundary versus that in the interior grows significantly with increasing dimensions [43], it is crucial to solve the equation by using only inner nodes and to use a fast approach to determine the Hamiltonian minimizing control.

To put the results in a general context we give an overview of some related work. For results on feedback control of infinite dimensional, second-order (in time), linear oscillators, see [21]; further we refer to the review article by Morris [39] on feedback control problems of PDEs considering in particular applications in the control of noise and plate vibrations. Moreover, we mention the monograph by Lasiecka and Triggiani [34] on control theory of PDEs. Regarding feedback stabilization of (the finite- and infinite-dimensional) Schrödinger equations, see, e.g., [7, 37, 38]. For optimal feedback control problems by solving HJB equations for reduced systems using proper orthogonal decomposition we refer to [1, 2, 22, 26, 32, 33], for reduced system using spectral elements to [31]. Regarding estimates for the error between the value functions of the continuous and semi-discrete optimal control problem in case of linear dynamics we refer to [16]. For publications on sparse grids we refer, e.g., to [11, 19, 23, 25, 43, 47]. For sparse approximation of PDEs in high dimensions, see, e.g., [14] and for sparse grids methods for solving Schrödinger equations we refer to [20, 24]. Higher order semi-Lagrangian schemes on sparse grids for second order HJB equations are considered in [46]. Furthermore, in [45] several two dimensional numerical examples for semi-Lagrangian schemes on sparse grids for first order HJB equation are considered. We further mention that model order reduction in combination with sparse grids was considered in [6, 42].

The paper is organized as follows. In Sect. 3 we formulate the closed-loop optimal control problems arising from control of the wave and Schrödinger equation, in Sect. 4 we recall the basic ideas of sparse grids, in Sect. 5 we formulate the semi-Lagrangian scheme, and in Sect. 6 we present several numerical examples illustrating our approach.

2 Notation

Throughout the paper we use for given interval (resp. bounded domain) $\Omega \subset \mathbb{K}$ with $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ the usual notation for the real-valued (resp. complex-valued) Lebesgue spaces $L^2(\Omega, \mathbb{K})$ (resp. $\bar{L}^2(\Omega, \mathbb{K})$), and analogue for Sobolev spaces $H_0^1(\Omega) = H_0^1(\Omega, \mathbb{R})$ (resp. $\bar{H}_0^1(\Omega) = H_0^1(\Omega, \mathbb{C})$), $m \in \mathbb{N}$. We set $H = L^2(\Omega, \mathbb{R})$ and $\bar{H} = L^2(\Omega, \mathbb{C})$. Furthermore, we introduce for a Hilbert space W the Bochner and Hölder spaces by $L^2(0, T; W)$ and $C^k(0, T; W)$, $k = 0, 1$, omitting the index for $k = 0$. For the Euclidean and maximum norm in \mathbb{R}^n , $n \in \mathbb{N}$, we use the usual notation $\|\cdot\|_2$ and $\|\cdot\|_\infty$.

3 Optimal Control Problems

In this section we introduce an optimal control problem for the wave and Schrödinger equation and the corresponding semi-discrete (in space) problems.

In the following, let $T > 0$ be a finite time horizon, $L > 0$, and c be a positive scalar.

3.1 Optimal Control of the Wave Equation

To formulate the optimal control problem for the wave equation, we first define the set of admissible controls

$$U_{\text{ad}}^w := L^2(0, T; U^w), \quad \text{with } U^w := \{u \in \mathbb{R}^m \mid u_a \leq u \leq u_b\} \quad (3.1)$$

for $u_a, u_b \in \mathbb{R}^m$, $m \in \mathbb{N}$. For controls $u \in U_{\text{ad}}^w$ the equation is given by

$$\begin{cases} \hat{y}_{tt} - c\Delta\hat{y} = B^w u & \text{in } (0, T) \times \Omega, \\ \hat{y}(0) = \hat{y}_0, \quad \hat{y}_t(0) = \hat{y}_1 & \text{in } \Omega, \\ \hat{y} = 0 & \text{on } (0, T) \times \partial\Omega \end{cases} \quad (3.2)$$

for initial state and velocity $\hat{y}_0 \in H_0^1(\Omega)$ and $\hat{y}_1 \in L^2(\Omega)$, and control operator $B^w(x) := (\sin(\pi x_1), \dots, \sin(m\pi x_2))$ for $x \in \Omega$. Equation (3.2) has a unique solution

$$\hat{y} \in C(0, T; H_0^1(\Omega)) \cap C^1(0, T; L^2(\Omega)), \quad (3.3)$$

see [35, pp. 275 and 288]. Let the cost functional be given by

$$J^w(u, \hat{y}) := \int_0^T (\beta_y \|\hat{y}(t) - y_d\|_H^2 + \beta_u |u(t)|^2) dt, \quad (3.4)$$

for $y_d \in H$ and $\beta_y > 0$ and $\beta_u > 0$. We denote the control-to-state operator for the wave equation by $\hat{y}_w[\cdot]$. The optimal control problem is given by

$$\min F^w(u) := J^w(u, \hat{y}_w[u]), \quad u \in U_{\text{ad}}^w. \quad (3.5)$$

The existence of a unique solution of the control problem (3.5) follows by classical arguments, see, e.g., [36].

Next, we introduce a semi-discrete formulation of the control problem, in particular we use the method of lines. For a given basis $b := (\varphi_1, \dots, \varphi_d)$, with $\varphi_i \in H_0^1(\Omega)$, $d \in \mathbb{N}$, we define

$$\begin{aligned} A &:= ((\nabla\varphi_i(x), \nabla\varphi_j(x))_{i,j=1,\dots,d}) && \text{(stiffness matrix),} \\ M &:= ((\varphi_i(x), \varphi_j(x))_{i,j=1,\dots,d}) && \text{(mass matrix).} \end{aligned} \quad (3.6)$$

Note that for our numerical experiments in Sect. 6 we will choose

$$\varphi_i(x) := \sin(i\pi x), \quad i = 1, \dots, d, \quad (3.7)$$

and obtain

$$A = \text{diag}((1/2(i\pi)^2)_{i=1,\dots,d}), \quad M = \text{diag}((1/2)_{i=1,\dots,d}). \quad (3.8)$$

although the following exposition holds for a general basis b .

A semi-discrete formulation of the wave equation with respect to a basis b is obtained by a first order system in time given by

$$\dot{y}_w(t) = f^w(y_w(t), u(t)), \quad t > 0, \quad y(0) = y_0 \quad (3.9)$$

with $y_w(t) \in \mathbb{R}^{2d}$ and the dynamics

$$f^w : \mathbb{R}^{2d} \times \mathbb{R}^m \rightarrow \mathbb{R}, \quad f^w(x, u) := \mathcal{A}^w x + \mathcal{B}^w u, \tag{3.10}$$

where

$$\mathcal{A}^w := \begin{pmatrix} 0 & I_d \\ -cM^{-1}A & 0 \end{pmatrix}, \quad \mathcal{B}^w := \begin{pmatrix} 0 \\ \text{id} \end{pmatrix}, \quad y_0 \in \mathbb{R}^{2d}. \tag{3.11}$$

Furthermore, we define the functional

$$J_h^w(u, y) := \int_0^T l^w(y(t), u(t)) dt \tag{3.12}$$

with running cost $l^w(x, u) := \beta_y x_1^T M x_1 + \beta_u u^T u$ for $x = (x_1, x_2) \in \mathbb{R}^{2d}, u \in U^w$. The semi-discrete optimal control problem is given as

$$\min F_h^w(u) := J_h^w(u, y_w[u]), \quad u \in U_{\text{ad}}^w. \tag{3.13}$$

Existence of a solution for (3.13) follows similarly as in the continuous case.

3.2 Optimal Control of the Schrödinger Equation

We define the set of admissible controls for the Schrödinger equation

$$U_{\text{ad}}^s := L^2(0, T; U^s), \quad \text{with } U^s := \{u \in \mathbb{R} \mid u_a \leq u \leq u_b\} \tag{3.14}$$

and introduce the equation for controls $u \in U_{\text{ad}}^s$ by

$$\begin{cases} i \hat{y}_t + c \Delta \hat{y} - u B^s \hat{y} = 0 & \text{in } (0, T) \times \Omega, \\ \hat{y}(0) = \hat{y}_0 & \text{in } \Omega, \\ \hat{y} = 0 & \text{on } (0, T) \times \partial\Omega, \end{cases} \tag{3.15}$$

with initial state $\hat{y}_0 \in \bar{H}_0^1(\Omega)$, and scalar $B^s \in \mathbb{R}$. Equation (3.15) has a unique solution in

$$C(0, T; \bar{H}_0^1(\Omega)) \cap C^1(0, T; \bar{H}^{-1}(\Omega)), \tag{3.16}$$

see [35]. We define the cost functional as

$$J^s(u, \hat{y}) := \int_0^T \left(\beta_y \|\hat{y}(t)\|_{\bar{H}}^2 - y_d \|\hat{y}(t)\|_{\bar{H}}^2 + \beta_u |u(t)|^2 \right) dt \tag{3.17}$$

with $y_d \in \bar{H}, \beta_y > 0$ and $\beta_u > 0$. The optimal control problem is given by

$$\min F^s(u) := J^s(u, \hat{y}_s[u]), \quad u \in U_{\text{ad}}^s \tag{3.18}$$

where, in analogy to the wave equation, $\hat{y}_s[u]$ denotes the solution of (3.15) for given $u \in U_{\text{ad}}^s$. The existence of a solution of (3.18) can be shown using classical arguments, see [3, Proposition 7.5].

We introduce a semi-discretization of the Schrödinger equation as a real-valued system as

$$f^s : \mathbb{R}^{2d} \times \mathbb{R} \rightarrow \mathbb{R}, \quad f^s(x, u) := \mathcal{A}^s x + u B^s y \tag{3.19}$$

with

$$\begin{aligned} \mathcal{A}^s &:= \begin{pmatrix} 0 & cM^{-1}A \\ -cM^{-1}A & 0 \end{pmatrix}, & \mathcal{M} &:= \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}, & \mathcal{B}^s &:= M^{-1}\hat{B}, \\ \hat{B} &:= \begin{pmatrix} 0 & -\bar{B} \\ \bar{B} & 0 \end{pmatrix}, & \bar{B} &= \text{diag}(B^s, \dots, B^s) \in \mathbb{R}^{d \times d}. \end{aligned} \tag{3.20}$$

The discrete cost functional is given by

$$J_h^s(u, y) := \int_0^T l^s(y(t), u(t)) dt \tag{3.21}$$

with running cost

$$l^s(x, u) := \beta_y x^T \mathcal{M}x + \beta_u u^T u, \tag{3.22}$$

where $x = (x_1, x_2) \in \mathbb{R}^{2d}$, $u \in U$. The semi-discrete optimal control problem is given as

$$\min F_h^s(u) := J_h^s(u, y_s[u]), \quad u \in U_{\text{ad}}^s. \tag{3.23}$$

Again, existence of a solution of (3.23) follows similarly as in the continuous case.

3.3 Suboptimal Feedback Control

We introduce the value function for both semi-discrete problems (3.13) and (3.23):

$$v(x, t) := \inf_{u \in U} \int_t^T l(y(s), u(s)) ds, \quad \text{s.t.} \tag{3.24}$$

$$\dot{y}(s) = f(y(s), u(s)), \quad y(t) = x, \quad s > t, \tag{3.25}$$

for $t \in [0, T]$ and $x \in \mathbb{R}^{2d}$ with $l = l^w$, $f = f^w$, and $U = U^w$ (resp. $l = l^s$, $f = f^s$, and $U = U^s$) for the wave (resp. Schrödinger) equation. The value function satisfies the dynamic programming principle

$$v(x, t) = \inf_{u \in U} \left(\int_t^\tau l(y(s), u(s)) ds + v(y(\tau), \tau) \right) \quad \text{in } \mathbb{R}^{2d} \times [0, T], \tag{3.26}$$

for all $\tau \in [t, T]$ and can be characterized as the unique viscosity solution of

$$\begin{cases} -v_t(x, t) + H(x, \nabla v) = 0 & \text{in } \mathbb{R}^{2d} \times [0, T], \\ v(x, T) = 0 & \text{in } \mathbb{R}^{2d}, \end{cases} \tag{3.27}$$

with Hamiltonian

$$H(x, p) := \sup_{u \in U} \left(-f(x, u)^T p - l(x, u) \right) \tag{3.28}$$

for $x \in \mathbb{R}^{2d}$ and $p \in \mathbb{R}^{2d}$, see [4, 29].

We assume that for the value function holds $v \in C^1(\mathbb{R}^{2d} \times [0, T])$. To derive a feedback control law from the value function we introduce the set-valued map (following the notation in [18, Chapter I]) given by

$$g^*(x, t) := \operatorname{argmin}_{u \in U} \left\{ -f(x, u)^T \nabla_x v(x, t) - l(x, u) \right\}. \tag{3.29}$$

We call $\underline{u} : \mathbb{R}^{2d} \times (t, T) \rightarrow U$ an admissible feedback control and define $u(s) := \underline{u}(y(s), s)$. Let \underline{u}^* be an admissible feedback control with

$$\underline{u}^*(x, s) \in g^*(x, s) \tag{3.30}$$

for all $(x, s) \in \mathbb{R}^{2d} \times (t, T)$ and admissible for the initial condition (x, t) , then we call \underline{u}^* an optimal feedback control and with the corresponding solution y^* we have

$$u^*(s) = \underline{u}^*(y^*(s), s). \tag{3.31}$$

We call the corresponding trajectory for given initial state (x, t) optimal.

Note that for f and l given as above we obtain from the feedback law in (3.29) that

$$\underline{u}^*(y^*(s), s) = P_{U_{ad}^w} \left(-\frac{1}{\beta_u} (\mathcal{B}^w)^T \nabla_x v(y^*(s), s) \right) \tag{3.32}$$

for the semi-discrete wave equation and

$$\underline{u}^*(y^*(s), s) = P_{U_{ad}^s} \left(-\frac{1}{\beta_u} y^*(s)^T (\mathcal{B}^s)^T \nabla_x v(y^*(s), s) \right) \tag{3.33}$$

for the semi-discrete Schrödinger equation, where $P_{U_{ad}^w}$ (resp. $P_{U_{ad}^s}$) denotes the projection on the set of admissible controls.

3.4 Curse of Dimensionality

Optimal control problems of type (3.24) and (3.25) allow to derive controls in feedback form from the corresponding value function given as the unique viscosity solution of an instationary HJB equation (3.27). However, when the problem arises from a semi-discrete optimal control problem governed from a PDE, it usually leads to a high dimensional state space and, because of the curse of dimensionality, the numerical approximation becomes very challenging.

For the numerical approach, the first step is a restriction of the state space to a suitable bounded subdomain $Q \subset \mathbb{R}^{2d}$. To make the problem numerically feasible and to reduce the computational effort two different aspects are crucial. On the one hand the reduction of the dimension of the underlying system and on the other hand efficient schemes for solving the HJB equation (3.27). In this paper we focus on the latter and analyze the approximation by adaptive semi-Lagrangian schemes on sparse grids. Nevertheless, a low dimensional approximation is necessary such that the problem is numerically feasible. Here, following the approach presented in [31], we consider semi-discretizations based on spectral elements and the aim is to control the lower modes (neglecting the behaviour of the higher modes). However, the presented framework could also be applied to optimal control problems of parabolic equations for which model order reduction techniques like proper orthogonal decomposition are very efficient (cf. the cited references in the introduction).

4 Sparse Grids

For the numerical approximation of the value function we use a semi-Lagrangian scheme on adaptive sparse grids. In the following we recall the main ideas behind (adaptive) sparse grids, more details can be found in e.g., [11, 19, 43]. Let here, to simplify the exposition, $Q = [0, 1]^d$, $d \in \mathbb{N}$. For a multiindex $\underline{l} = (l_1, \dots, l_d) \in \mathbb{N}^d$ we introduce a mesh $Q_{\underline{l}}$ with

mesh parameters $h_{\underline{l}} = (h_{l_1}, \dots, h_{l_d})$ which are constant $h_{l_i} := 2^{-l_i}$ in each direction but may differ with respect to the dimensions. The grid points are denoted by

$$x_{\underline{l}, \underline{j}} = (x_{l_1, j_1}, \dots, x_{l_d, j_d}), \quad x_{l_t, j_t} = j_t \cdot h_{l_t}, \quad t = 1, \dots, d. \tag{4.1}$$

Here, \underline{l} denotes the level which characterizes the resolution of the discretization and \underline{j} defines the position of the mesh point. Let

$$V_{\underline{l}} := \text{span} \left\{ \varphi_{\underline{l}, \underline{j}} \mid j_t = 0, \dots, 2^{l_t}, t = 1, \dots, d \right\} \tag{4.2}$$

be the space of all d -dimensional piecewise d -linear hat functions. The hierarchical difference space is given by

$$W_{\underline{l}} := V_{\underline{l}} \setminus \bigoplus_{t=1}^d V_{\underline{l}-\underline{e}_t}, \tag{4.3}$$

where \underline{e}_t denotes the t th unit vector. Therefore $W_{\underline{l}}$ consists of the functions in $V_{\underline{l}}$ which are not in any $V_{\underline{l}-\underline{e}_t}$ and this allows the construction of a multilevel subspace splitting. Thus we define, denoting by \leq a component-wise relation here and in the following,

$$V_{\underline{l}} = \bigoplus_{\underline{k} \leq \underline{l}} W_{\underline{k}}, \tag{4.4}$$

cf. Fig. 1. With $V_n := V_{(n, \dots, n)}$ for $n \in \mathbb{N}$ every $f \in V_n$ can be characterized as

$$f(\underline{x}) = \sum_{|\underline{l}|_{\infty} \leq n} \sum_{\underline{j} \in B_{\underline{l}}} \alpha_{\underline{l}, \underline{j}} \cdot \varphi_{\underline{l}, \underline{j}}(\underline{x}), \tag{4.5}$$

with so-called hierarchical coefficients $\alpha_{\underline{l}, \underline{j}} \in \mathbb{R}$ and for

$$B_{\underline{l}} := \left\{ \underline{j} \in \mathbb{N}^d \mid \begin{array}{l} j_t = 1, \dots, 2^{l_t} - 1, \text{ } j_t \text{ odd}, \text{ } t = 1, \dots, d, \text{ if } l_t > 1, \\ j_t = 0, 1, 2, \text{ } t = 1, \dots, d, \text{ if } l_t = 1 \end{array} \right\}. \tag{4.6}$$

In nodal (4.2) or hierarchical (4.5) basis a function $f \in V_n$ with discretization level n is characterized by $(2^n + 1)^d$ points.

To proceed, we need the so-called Sobolev-space with dominating mixed derivative H^2_{mix} . We introduce the corresponding norm and semi-norm

$$\|f\|_{H^2_{\text{mix}}(Q)}^2 = \sum_{0 \leq k \leq 2} |\partial_{x_1^{k_1} \dots x_d^{k_d}} f|_2^2, \quad \text{and} \quad |f|_{H^2_{\text{mix}}(Q)} = \left\| \partial_{x_1^2 \dots x_d^2} f \right\|_2, \tag{4.7}$$

respectively. One formally defines

$$H^2_{\text{mix}}(Q) = \left\{ f \in H \mid \|f\|_{H^2_{\text{mix}}} \leq C \text{ for } C > 0 \right\} \tag{4.8}$$

which satisfies the relation $H^{2-d}(Q) \subset H^2_{\text{mix}}(Q) \subset H^2(Q)$, see [25]. For $f \in H^2_{\text{mix}}(Q)$ there holds, see e.g., [11],

$$\|f_{\underline{l}}\|_2 \leq C(d) \cdot 2^{-2|\underline{l}|_1} |f|_{H^2_{\text{mix}}(Q)} \tag{4.9}$$

with constant $C(d) > 0$ depending on the dimension d and

$$f_{\underline{l}} := \sum_{\underline{j} \in B_{\underline{l}}} \alpha_{\underline{l}, \underline{j}} \varphi_{\underline{l}, \underline{j}}(x) \in W_{\underline{l}}.$$

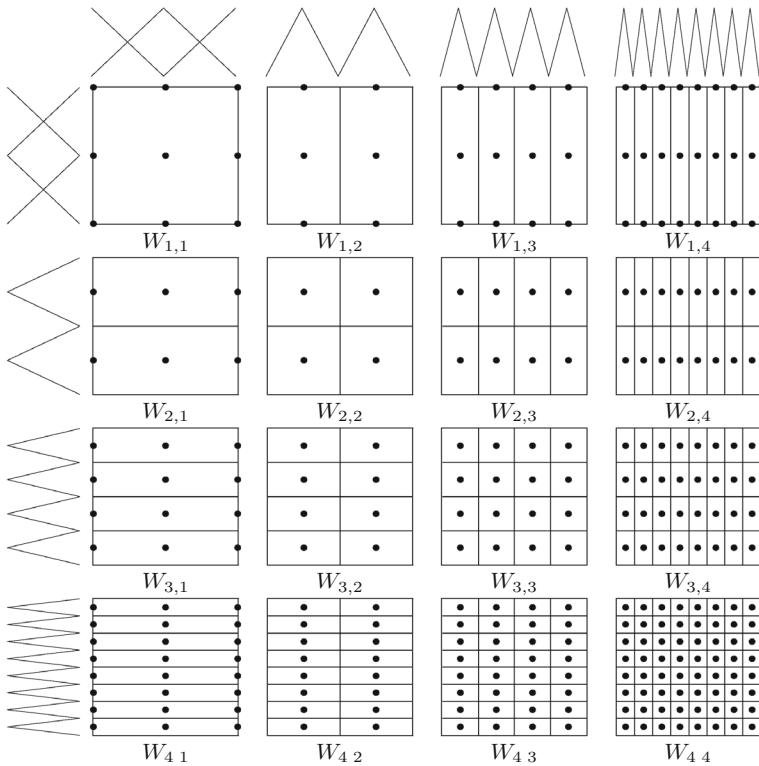


Fig. 1 Supports of the basis functions $\varphi_{l,j}$ of the hierarchical subspaces W_l of the space $V_4 = V_{(4,4)}$, where the functions of each W_l have disjoint support. The regular sparse grid V_4^s contains the upper left triangle of spaces

The hierarchical mesh becomes a sparse mesh when taking out those basis function which have a small contribution to the representation of the function, i.e. those with a small support due to estimate (4.9). Following Griebel [23] and Zenger [47], we replace $\|L\|_\infty \leq n$ by the rule

$$\|L\|_1 \leq n + d - 1. \tag{4.10}$$

For the dimension of the sparse grid space it holds $\dim V_n^s = \mathcal{O}(2^n \cdot n^{d-1})$, in comparison to regular grids where we have $\dim V_n = \mathcal{O}(2^{nd})$. For functions $f \in H_{\text{mix}}^2(Q)$ there holds the error estimate

$$\|f - f_n^s\|_2 = \mathcal{O}\left(h_n^2 (\log h_n^{-1})^{d-1}\right) \tag{4.11}$$

in comparison to the approximation on regular grids, where we have

$$\|f - f_n^s\|_2 = \mathcal{O}(h_n^2). \tag{4.12}$$

This fact leads to a strong reduction of the computational storage consumption in comparison to a full mesh approach for a similar approximation quality. If the required H_{mix}^2 -regularity is given, the points in the sparse grid are optimal in a cost-benefit analysis [11].

4.1 Adaptive Sparse Grids

For a further reduction of the number of nodes we use an adaptive refinement strategy as in [10], where the values of the hierarchical coefficients are employed as error indicators [17, 43].

We now collect the indices $(\underline{l}, \underline{j})$ of the employed adaptive sparse grid functions in an index set \mathcal{I} , denote the resulting discretization mesh as $Q_{\mathcal{I}}$ and start with some suitable initialization $\mathcal{I} = \{(\underline{l}, \underline{j}) | \varphi_{\underline{l}, \underline{j}} \in V_n^s\}$ for a small n . In the iterative adaptive procedure to build the refined index set \mathcal{I} , an index $(\underline{l}, \underline{j}) \in \mathcal{I}$ is then marked for refinement if there holds, for given parameter $\varepsilon > 0$,

$$|a_{\underline{l}, \underline{j}}| \|\phi_{\underline{l}, \underline{j}}\| > \varepsilon \tag{4.13}$$

for the hierarchical coefficient $a_{\underline{l}, \underline{j}}$ in the function representation (4.5). In such a case the $2d$ children, left and right in each dimension, are added to \mathcal{I} , where for consistency one might need to add fathers in other dimensions of these newly added sparse grid points. The spatially adaptive refinement algorithm to build an adaptive sparse grid for given function F , initial \mathcal{I} , and ε is presented in Algorithm 1 and is based on [10] and described in more detail therein.

Algorithm 1: Spatially adaptive refinement

Data: initial \mathcal{I} , refinement threshold ε and function evaluation F
Result: refined \mathcal{I} , adaptive sparse grid approximation of F in $V_{\mathcal{I}}$

for all indices $(\underline{l}, \underline{j}) \in \mathcal{I}$ **do**
 | compute $F(\underline{x}_{\underline{l}, \underline{j}})$ ▷ evaluate F on initial grid
compute hierarchical values $\alpha_{\underline{l}, \underline{j}}$ for all indices;

while indices are added to \mathcal{I} **do**
 for $(\underline{l}, \underline{j}) \in \mathcal{I}$ **do** ▷ look at all indices
 | **if** $|\alpha_{\underline{l}, \underline{j}}| \cdot \|\phi_{\underline{l}, \underline{j}}\| > \varepsilon$ **then**
 | **for** $t = 1, \dots, d$ **do** ▷ surplus is large
 | **if** $(\tilde{\underline{l}}, \tilde{\underline{j}}) \notin \mathcal{I}$ for $\tilde{\underline{l}} = \underline{l} + e_t$ and $\tilde{\underline{j}} \in \{j + j_t e_t \pm 1\}$ **then**
 | $\mathcal{I} = \mathcal{I} \cup (\tilde{\underline{l}}, \tilde{\underline{j}})$ ▷ add children not in \mathcal{I}
check $\forall (\underline{l}, \underline{j}) \in \mathcal{I}: (\tilde{\underline{l}}, \tilde{\underline{j}}) \in \mathcal{I}$ for $\tilde{\underline{l}} \leq \underline{l}$ and $\text{supp}(\phi_{\tilde{\underline{l}}, \tilde{\underline{j}}}) \cap \text{supp}(\phi_{\underline{l}, \underline{j}}) \neq \emptyset$;

for all added indices $(\underline{l}, \underline{j}) \in \mathcal{I}$ **do**
 | compute $F(\underline{x}_{\underline{l}, \underline{j}})$ ▷ evaluate F at new grid points
compute hierarchical values $\alpha_{\underline{l}, \underline{j}}$ for newly added indices

Additionally, we also use coarsening of the spatially adaptive sparse grid, i.e. for given parameter $\eta > 0$ we remove an index $(\underline{l}, \underline{j})$ from \mathcal{I} if

$$|a_{\underline{l}, \underline{j}}| \|\phi_{\underline{l}, \underline{j}}\| < \eta, \tag{4.14}$$

and no children of $(\underline{l}, \underline{j})$ are in \mathcal{I} , see Algorithm 2. This is to avoid unnecessary function evaluations on sparse grid points whose basis function only have little contribution, again see [10] for more details. A typical choice is $\eta = \varepsilon/10$, which we will use in our experiments in Sect. 6.

Algorithm 2: Spatially adaptive coarsening

Data: index set \mathcal{I} , coarsening threshold η , and $\alpha_{\underline{l}, \underline{j}} \forall (\underline{l}, \underline{j}) \in \mathcal{I}$
Result: coarsened index set \mathcal{I}
while indices are removed from \mathcal{I} **do**
 for $(\underline{l}, \underline{j}) \in \mathcal{I}$ **do** ▷ look at all indices
 if $|\alpha_{\underline{l}, \underline{j}}| \cdot \|\phi_{\underline{l}, \underline{j}}\| < \eta$ **then** ▷ surplus is small
 if $\forall t = 1, \dots, d: (\tilde{\underline{l}}, \tilde{\underline{j}}) \notin \mathcal{I}$ for $\tilde{\underline{l}} = \underline{l} + \underline{e}_t$ and $\tilde{\underline{j}} \in \{\underline{j} + j_t \underline{e}_t \pm 1\}$ **then**
 $\mathcal{I} = \mathcal{I} \setminus (\underline{l}, \underline{j})$ ▷ remove if no children in \mathcal{I}

Note that several norms are possible in (4.13) and (4.14), e.g., $L^\infty(Q)$, $L^2(Q)$, $L^1(Q)$, $H^1(Q)$ or mixtures thereof. In the numerical experiments we use $\|\cdot\|_{L^\infty(Q)}$ which is one for the employed basis functions.

5 The Semi-Lagrangian Scheme

To compute the value function numerically we apply a semi-Lagrangian scheme, cf., e.g., [15]. For a bounded domain $Q \subset \mathbb{R}^{2d}$ we apply for its discretization $Q_{\mathcal{I}}$ the procedure

$$\begin{cases} v^k(\underline{x}) = \min_{u \in U} (\Delta t l(\underline{x}, u) + I[v^{k+1}](y_{\underline{x}}(\Delta t))), \\ v^k(\underline{x}) = 0 \end{cases} \tag{5.1}$$

for all $\underline{x} \in Q_{\mathcal{I}}$, time step $\Delta t > 0$, $k = K, \dots, 0$, and $K = T/\Delta t$, where \mathcal{I} is either the index set of a regular or adaptive sparse grid. The interpolation operator I is defined on the grid points of $Q_{\mathcal{I}}$ by $I[v](\underline{x}) = v(\underline{x})$ for all $\underline{x} \in Q_{\mathcal{I}}$ and $y_{\underline{x}}(\Delta t)$ denotes the state obtained by a time discretization scheme when going from \underline{x} one step forward in time.

5.1 Computational Domain and Boundary Treatment

When using sparse grids, the number of points on the boundary increases, in comparison to inner points, strongly with respect to the refinement level and the dimension, see, e.g., [43]. To avoid this behaviour we use so-called fold out ansatz functions which are defined with inner nodes only and extrapolate them to and over the boundary following ideas developed in [10, 43], see Fig. 2a, b.

Besides this, we have to prescribe some boundary condition for solving the dynamic programming equation on the extended domain. We set the second derivative equal to zero, which corresponds to a linear extension of the values on inner nodes over the boundary. With the fold out basis functions we have this extrapolation outside of the domain naturally, see Fig. 2c. For the normal basis function we set the value of v for an evaluation point outside of the domain, i.e. $(\tilde{x}, y) = (x \pm h, y)$ with $(x, y) \in \partial Q_I$ and $h > 0$ the distance from \tilde{x} to Q_I , as

$$v(\tilde{x}, y) = v(x \pm h, y) := 2 \cdot v(x, y) \mp v(x \mp h, y), \tag{5.2}$$

taking either the backward or forward difference quotient depending on which part of the boundary we are. If the evaluation point is outside of the domain in more than one dimension we treat all affected dimensions in this way.

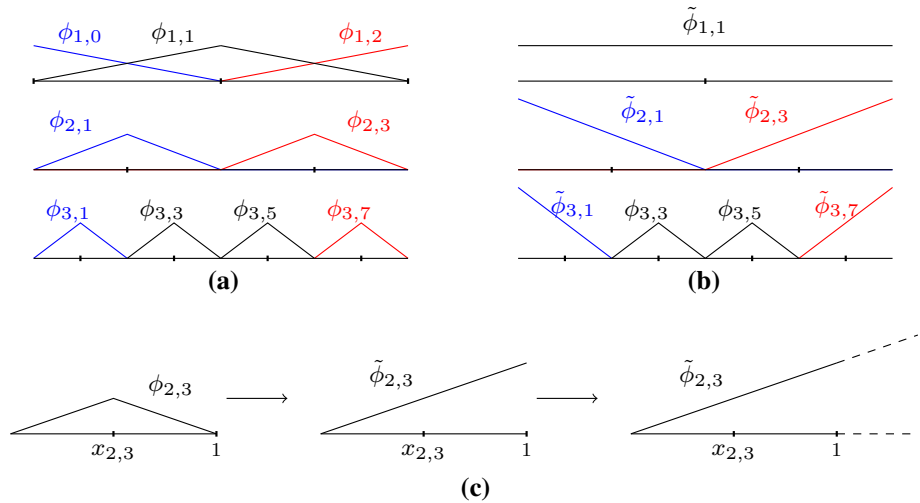


Fig. 2 Normal and fold out basis functions. **a** Normal basis, **b** fold-out basis, **c** last basis before the boundary gets folded up and linearly extrapolated across the boundary

5.2 Computational Aspects of the Minimization in the Hamiltonian and the Feedback Law

In general the determination of the minimum in the right hand side of (5.1) over the set of admissible controls on sparse grids is a non-trivial task; already on regular grids this questions has to be addressed carefully, for a discussion of first-order and second-order algorithms see [30]. For global minimization algorithms on sparse grids see, e.g., [40]. In the following we discuss two different methods to determine the minimizing control which can be used within the SL-scheme as well as for computing optimal trajectories.

5.2.1 Feedback Law Based on Minimization by Comparison

In this approach a finite subset $U_\sigma \subset U$ can be chosen over which the minimizer is determined by comparison over its elements, i.e. we chose

$$\underline{u}^*(\underline{x}, s) \in \operatorname{argmin}_{u \in U} \left(v^k(\underline{x} + \Delta t \cdot f(\underline{x}, u), s) + \Delta t \cdot l(\underline{x}, u) \right), \quad \underline{x} \in Q_{\mathcal{I}}, \quad (5.3)$$

see, e.g., [4, 15]. This approach is easy to implement and allows to consider also non-smooth cost functionals, however it is very time consuming in higher dimensions in particular if the control has several components.

5.2.2 Feedback Law Based on the Gradient of the Value Function

In case of differentiability of the value function a projection formula for the minimization using the gradient of the value function can be used, see (3.32) and (3.33). Within the semi-Lagrangian scheme, the value function of the previous time step is used to evaluate the right hand side in the feedback law. A suitable approximation of the gradient has to be chosen.

A finite difference approximation using the interpolation on sparse grids can be used given by

$$u^n(\underline{x}, s) = P_U \left(-\frac{1}{\beta_u} (\mathcal{B}^w)^T \nabla^h v^{n+1}(\underline{x}, s) \right) \tag{5.4}$$

for the wave equation and

$$u^n(\underline{x}, s) = P_U \left(-\frac{1}{\beta_u} \underline{x}^T (\mathcal{B}^s)^T \nabla^h v^{n+1}(\underline{x}, s) \right) \tag{5.5}$$

for the Schrödinger equation, respectively, with

$$\left(\nabla^h v^{n+1}(\underline{x}, s) \right)_i := \frac{v^{n+1}(\underline{x} + h_i, s) - v^{n+1}(\underline{x} - h_i, s)}{2h} \tag{5.6}$$

for $h_i = (0, \dots, h, 0, \dots, 0) \in \mathbb{R}^{2d}$ and $h > 0$. However, the choice of $h > 0$ in the difference quotient approximation is not naturally given on sparse grids as it is the case for regular grids.

In Sect. 6 we will compare numerically both approaches for determining the minimizing control.

5.3 Evaluation of the Right Hand Side in (5.1)

For the computation of $y_{\underline{x}}(\Delta t)$ we use the second order Heun scheme

$$y_{\underline{x}}(\Delta t) := \frac{1}{2} (\underline{x} + \tilde{\underline{x}} + \Delta t \cdot f(\tilde{\underline{x}}, u)), \tag{5.7}$$

where $\tilde{\underline{x}}$ is computed with the Euler scheme

$$y_{\underline{x}}(\Delta t) := \underline{x} + \Delta t f(\underline{x}, u). \tag{5.8}$$

For the evaluation of the right hand side of (5.1) in the n -dimensional space given by

$$F_k(\underline{x}) = \min_{u \in U} (v_k(y_{\underline{x}}(\Delta t)) + \Delta t l(\underline{x}, u)), \quad k = K, \dots, 0, \tag{5.9}$$

we interpolate the expression on sparse grids as presented in Algorithm 3.

Algorithm 3: Adaptive SL-SG scheme

Data: suitable initial index set \mathcal{I} , refinement constant ε and coarsening constant η

Result: sequence of adaptive sparse grid solutions $v_k \in V_{\mathcal{I}(k)}$ for $k = K, \dots, 0$

Alg. 1 with \mathcal{I}, ε , and $F_k(\underline{x}) = 0$ ▷ interpolate φ by $v_0 \in V_{\mathcal{I}(0)}$

Alg. 2 with $\mathcal{I}(0), \eta$, and v_0 ▷ coarsen v_0

for $k = K - 1, \dots, 0$ **do** ▷ iterate in time with $\Delta t = T/K$

Alg. 1 with $\mathcal{I}(k), \varepsilon$ and $F_k(\underline{x})$ ▷ compute $v_{k-1} \in V_{\mathcal{I}(k-1)}$

Alg. 2 with $\mathcal{I}(k - 1), \eta$ and v_{k-1} ▷ coarsen $v_{k-1} \in V_{\mathcal{I}(k-1)}$

5.4 Computation of the Trajectory

For the computation of optimal trajectories we solve the dynamical system (3.25) with the control given by the feedback laws using comparison after (5.3) or using the gradient of the value function after (5.4) and (5.5), respectively. The dynamical system is solved by the Heun time-stepping scheme. Note that one can choose the stepsize h for computing (5.6) for the trajectory differently from the choice within the semi-Lagrangian scheme.

The computational domain is chosen (by ‘testing’ and ‘checking’) in such a way that the trajectory does not reach the boundary to avoid numerical artefacts.

6 Numerical Examples

In this section we present several numerical examples in which we study for optimal control problems of type (3.24) and (3.25) the accuracy of the discrete value function (for $d = 2$) and optimal trajectories (for $2 \leq d \leq 8$) when solving the corresponding HJB equation by the SL-SG-scheme described in Sect. 5. For the study of convergence of the SL-SG-scheme we focus on the discretization error in space while using a time resolution which is “good enough”. While on regular grids various error estimates for semi-Lagrangian schemes are known, see, e.g., [4, Appendix A] and [5], on sparse grids very little results are available in the literature, see, e.g., [46] where under a certain assumptions an estimate is derived.

Since we only have an inefficient proof-of-concept sparse grid implementation available, we abstain from giving runtimes. Using efficient sparse grid implementations [12,43] would significantly (an order of magnitude or more) change the needed runtime.

In the following experiments with the semi Lagrangian scheme, we use for the feedback law by comparison (5.3) a discretization of the control space with 40 equidistantly spaced controls in one dimension, while for the feedback law by using the gradient after (5.4) and (5.5) we use the stepsize $h = \frac{1}{40}$.

6.1 Reference Solutions

To analyze the accuracy of our computed discrete value function we compare it with a reference solution v_{ref} computed with a higher order finite difference code on a uniform mesh based on methods developed in [9,41]. We compute a reference solution v_{ref} by an ENO scheme as a variant of a Lax–Friedrichs scheme

$$v_{\text{ref}}^{k-1}(x_I) = v_{\text{ref}}^k(x_I) - \Delta t \mathcal{H}_{LF} \left(x_I, D^+ v_{\text{ref}}^k(x_I), D^- v_{\text{ref}}^k(x_I) \right), \tag{6.1}$$

where \mathcal{H}_{LF} is the numerical Hamiltonian, $D^\pm v_{\text{ref}}^k(x_I)$ are higher order approximations of the gradient in grid point x_I , $I \in \mathbb{Z}$, k is the time step, Δt the temporal mesh parameter, and coupled with a Runge–Kutta time discretization scheme of second order, for details see [31]. For the numerical realization of the scheme we use the software library ROC-HJ (see [8]).

To quantify the error in the optimal trajectories for the control problem for the semi-discrete wave equation we compute reference trajectories y_r in state space and u_r in control space using a Riccati approach, i.e. we solve backward in time

$$\begin{cases} -P_t(t) = \mathcal{A}P(t) - P(t)\mathcal{A} + P^T(t)\mathcal{B}^T \mathcal{R} \mathcal{B} P(t) = 0, & t > 0 \\ P(T) = 0 \end{cases} \tag{6.2}$$

with $\mathcal{A} = \mathcal{A}^w$, $\mathcal{B} = \mathcal{B}^w$ and set the feedback operator as

$$\underline{u}(x, t) = -\mathcal{R}^{-1} \mathcal{B}^T P(t)x \tag{6.3}$$

with $\mathcal{R} = \beta_u$. For computing the trajectory from our solution, we employ $h = \frac{2}{40}$ in (5.6). This is to decouple the step sizes for the semi-Lagrangian scheme and the trajectory computation.

We estimate the error in the value function by computing the vector difference to the reference solution on its discretization grid, and denote it by

$$e_2^v := \|v - v^*\|_{L^2(Q)} \approx \frac{\|v - v_r\|_2}{\sqrt{N_r}}, \tag{6.4}$$

$$e_\infty^v := \|v - v^*\|_{L^\infty(Q)} \approx \|v - v_r\|_\infty. \tag{6.5}$$

For the errors in the state and control we use the following notation:

$$e_2^y := \|y - y^*\|_{L^2(0,T;\mathbb{R}^{2d})}, \quad e_\infty^y := \|y - y^*\|_{L^\infty(0,T;\mathbb{R}^{2d})}, \tag{6.6}$$

$$e_2^u := \|u - u^*\|_{L^2(0,T;\mathbb{R}^m)}, \quad e_\infty^u := \|u - u^*\|_{L^\infty(0,T;\mathbb{R}^m)}, \tag{6.7}$$

and approximate the L^∞ -error by $\max_{k=1,\dots,T/\Delta t} |y^k - y_r^k|$ and the L^2 -error by

$$\frac{1}{\sqrt{T/\Delta t}} \sum_{k=1}^{T/\Delta t} |y^k - y_r^k|^2.$$

Furthermore, we give convergence rates for the different discretization errors e_i as

$$\rho_e(i) = \log_2 \left(\frac{e_{i-1}}{e_i} \right), \tag{6.8}$$

where e_i denotes the error on level i or the error for $\varepsilon = 0.1 \times 1/2^i$, respectively, and e is any one of the above errors.

6.2 Control of the Harmonic Oscillator

As a first example we consider a simplification of the semi-discrete wave equation, namely the Harmonic oscillator, i.e. we consider dynamics of type (3.10) with

$$\beta_y = 2, \quad \beta_u = 0.1, \quad T = 1, \quad \Delta t = 0.01, \quad \mathcal{A}^w = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \mathcal{B}^w = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{6.9}$$

initial data $x \in \mathbb{R}^2$, computational domain $Q = [-1, 1]^2$, $U_w = [-3.5, 3.5]$. We solve the corresponding HJB equation (3.27) and compare different variants of our semi-Lagrangian scheme on sparse grids. The reference value function is computed as described in Sect. 6.1 on the domain $[-1, 1]^2$ with $\Delta x = 1/200 \times \text{length}(Q) = 1/200 \times 2$ and $\Delta t = T/1000$.

6.2.1 Error in the Value Function

To show the overall behaviour of the four different variants of our approach (namely using normal or fold out basis functions and uniform or adaptive refinement, respectively) we present in Fig. 3 combined convergence plots for the error in the value function at the initial time in comparison to the reference solution.

We also show two exemplary convergence tables with more detailed numbers. In Table 1 we give the error on regular sparse grids using the scheme presented in Algorithm 3 which includes nodes on the boundary, as well as fold out basis functions following Sect. 5.1; while in Table 2 we compare the error on the adaptive sparse grids with normal hat functions for decreasing refinement thresholds at the initial time zero by using either the gradient or comparison of all actions to realize the minimization on the sparse grid.

From Table 1 and Fig. 3a we can see that the different variants show the similar convergence of the error in regard to the level. Since a regular sparse grid with the fold-out basis functions

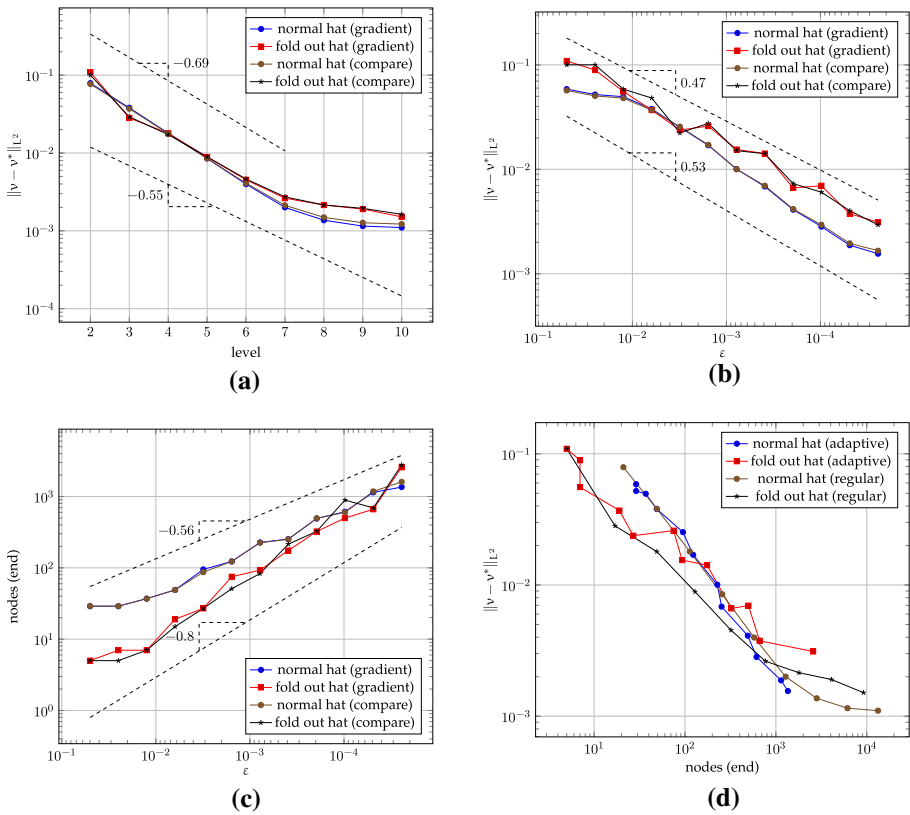


Fig. 3 Harmonic oscillator (6.9) in 2D: error in the approximating value function comparing the scheme based on sparse grids using normal hat and fold out basis functions and feedback laws from 5.2.1 and 5.2.2. **a** Error in the value function versus level, **b** error in the value function versus ϵ , **c** nodes versus ϵ and **d** error in the value function versus nodes, feedback law based on gradient approach

Table 1 Harmonic oscillator (6.9) in 2D: error in the value function with regular sparse grids using both kinds of basis functions and the gradient approach

Level	Normal				Fold out			
	e_2^v	ρe_2^v	e_∞^v	ρe_∞^v	e_2^v	ρe_2^v	e_∞^v	ρe_∞^v
2	7.91×10^{-2}	–	1.09×10^{-1}	–	1.09×10^{-1}	–	2.68×10^{-1}	–
3	3.81×10^{-2}	1.05	5.29×10^{-2}	1.05	2.82×10^{-2}	1.95	4.02×10^{-2}	2.74
4	1.80×10^{-2}	1.08	2.66×10^{-2}	0.99	1.80×10^{-2}	0.65	4.19×10^{-2}	–0.06
5	8.50×10^{-3}	1.08	1.44×10^{-2}	0.89	8.89×10^{-3}	1.02	3.07×10^{-2}	0.45
6	3.98×10^{-3}	1.09	8.72×10^{-3}	0.73	4.53×10^{-3}	0.97	2.42×10^{-2}	0.34
7	2.00×10^{-3}	0.99	5.93×10^{-3}	0.56	2.62×10^{-3}	0.79	2.04×10^{-2}	0.25
8	1.37×10^{-3}	0.55	5.75×10^{-3}	0.04	2.15×10^{-3}	0.29	2.06×10^{-2}	–0.02
9	1.15×10^{-3}	0.25	3.84×10^{-3}	0.58	1.90×10^{-3}	0.17	1.89×10^{-2}	0.13
10	1.10×10^{-3}	0.06	3.45×10^{-3}	0.16	1.51×10^{-3}	0.33	1.25×10^{-2}	0.60

Table 2 Harmonic oscillator (6.9) in 2D: error in the value function for adaptively refined sparse grids using compare and gradient approaches with the normal basis functions

ε	Gradient				Compare			
	e_2^v	ρe_2^v	e_∞^v	ρe_∞^v	e_2^v	ρe_2^v	e_∞^v	ρe_∞^v
5.00 ₋₂	5.86×10^{-2}		7.54×10^{-2}		5.69×10^{-2}		7.45×10^{-2}	
2.50 ₋₂	5.20×10^{-2}	0.17	6.44×10^{-2}	0.23	5.05×10^{-2}	0.17	6.31×10^{-2}	0.24
1.25 ₋₂	4.96×10^{-2}	0.07	6.18×10^{-2}	0.06	4.81×10^{-2}	0.07	5.95×10^{-2}	0.08
6.25 ₋₃	3.79×10^{-2}	0.39	5.13×10^{-2}	0.27	3.69×10^{-2}	0.38	4.95×10^{-2}	0.27
3.13 ₋₃	2.53×10^{-2}	0.58	3.73×10^{-2}	0.46	2.56×10^{-2}	0.53	3.86×10^{-2}	0.36
1.56 ₋₃	1.70×10^{-2}	0.57	2.51×10^{-2}	0.57	1.71×10^{-2}	0.58	2.53×10^{-2}	0.61
7.81 ₋₄	1.01×10^{-2}	0.76	1.63×10^{-2}	0.62	1.00×10^{-2}	0.78	1.60×10^{-2}	0.65
3.91 ₋₄	6.84×10^{-3}	0.56	1.03×10^{-2}	0.66	6.97×10^{-3}	0.52	1.04×10^{-2}	0.63
1.95 ₋₄	4.11×10^{-3}	0.74	6.40×10^{-3}	0.68	4.16×10^{-3}	0.74	6.46×10^{-3}	0.68
9.77 ₋₅	2.83×10^{-3}	0.54	4.51×10^{-3}	0.50	2.94×10^{-3}	0.50	4.61×10^{-3}	0.49
4.88 ₋₅	1.88×10^{-3}	0.59	3.45×10^{-3}	0.39	1.95×10^{-3}	0.59	3.92×10^{-3}	0.23
2.44 ₋₅	1.56×10^{-3}	0.27	5.31×10^{-3}	-0.62	1.66×10^{-3}	0.23	5.60×10^{-3}	-0.51

uses less points, it is the preferred choice in this setting, although for very fine resolution the convergence stagnates. Note that for a sparse grid with level 10 one has 13,313 nodes with normal hat functions and 9217 with fold out hat functions, respectively. To study the influence of the artificial boundary condition on the error, we also computed regular sparse grid solutions on the enlarged domain $[-2, 2]^2$, but evaluated the error only on $Q = [-1, 1]^2$. We observe overall very similar convergence behaviour for this setting and essentially identical errors for the normal and fold out basis functions, which indicates that the treatment of the boundary of the enlarged domain has little effect on the value function in Q .

From Table 2 and Fig. 3b we also see no big difference between the compare and gradient variants. We observe that the normal hat functions show slightly better performance when looking at the decrease of the error in comparison to ε . When taking the effort into account by counting the number of points in the adaptive sparse grids at the end of the computation, we see a stronger increase in the number of points for the fold-out basis function than for the normal ones, while for coarser ε the fold-out ones use less basis functions, see Fig. 3c. When comparing the error reduction with increasing number of nodes for adaptive versus regular grids, we see in Fig. 3d, that for normal hat functions there is no real difference, while for fold out hat functions the regular approach performs better and more stable.

6.2.2 Error in the Trajectory and Control

Next we study for given initial point $x = (0.4, -0.2)^T$ the accuracy of the corresponding trajectory and control when using regular/adaptive sparse grids, the gradient/comparison approach, and normal/fold out basis functions. Overall, the behaviour is very similar over the different measured errors, in particular when comparing the different basis functions and feedback laws on regular sparse grids. However, on adaptive sparse grids we observe for the fold out basis functions oscillating convergence behaviour.

Table 3 Harmonic oscillator (6.9) in 2D: error in the trajectory and control with regular sparse grids using the gradient approach and fold out basis functions

Level	e_2^y	$\rho_{e_2^y}$	e_∞^y	$\rho_{e_\infty^y}$	e_2^u	$\rho_{e_2^u}$	e_∞^u	$\rho_{e_\infty^u}$
2	4.17×10^{-1}	–	3.60×10^1	–	7.31×10^{-1}	–	1.68×10^0	–
3	7.64×10^{-2}	2.45	6.53×10^0	2.46	1.33×10^{-1}	2.46	2.74×10^{-1}	2.61
4	3.64×10^{-2}	1.07	3.10×10^0	1.07	6.09×10^{-2}	1.12	1.18×10^{-1}	1.21
5	1.90×10^{-2}	0.93	1.62×10^0	0.93	3.18×10^{-2}	0.94	5.97×10^{-2}	0.99
6	1.03×10^{-2}	0.89	8.77×10^{-1}	0.89	1.72×10^{-2}	0.89	3.20×10^{-2}	0.90
7	5.88×10^{-3}	0.80	5.04×10^{-1}	0.80	9.91×10^{-3}	0.79	2.02×10^{-2}	0.66
8	3.60×10^{-3}	0.71	3.09×10^{-1}	0.70	6.09×10^{-3}	0.70	1.24×10^{-2}	0.70
9	4.47×10^{-3}	–0.31	3.80×10^{-1}	–0.30	7.39×10^{-3}	–0.28	1.21×10^{-2}	0.03
10	4.08×10^{-3}	0.13	3.49×10^{-1}	0.13	6.80×10^{-3}	0.12	1.25×10^{-2}	–0.04

Table 4 Harmonic oscillator (6.9) in 2D: error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions

ε	e_2^y	$\rho_{e_2^y}$	e_∞^y	$\rho_{e_\infty^y}$	e_2^u	$\rho_{e_2^u}$	e_∞^u	$\rho_{e_\infty^u}$
5.00 _{–2}	4.17×10^{-1}	–	3.60×10^1	–	7.31×10^{-1}	–	1.68×10^0	–
2.50 _{–2}	3.99×10^{-1}	0.07	3.43×10^1	0.07	6.89×10^{-1}	0.09	1.34×10^0	0.33
1.25 _{–2}	1.92×10^{-1}	1.05	1.63×10^1	1.07	3.15×10^{-1}	1.13	5.11×10^{-1}	1.39
6.25 _{–3}	7.40×10^{-2}	1.38	6.10×10^0	1.42	1.24×10^{-1}	1.34	2.02×10^{-1}	1.34
3.13 _{–3}	3.24×10^{-2}	1.19	2.79×10^0	1.13	6.01×10^{-2}	1.04	1.49×10^{-1}	0.44
1.56 _{–3}	5.99×10^{-2}	–0.89	5.17×10^0	–0.89	1.09×10^{-1}	–0.86	2.36×10^{-1}	–0.66
7.81 _{–4}	2.34×10^{-2}	1.35	2.03×10^0	1.35	4.53×10^{-2}	1.27	9.95×10^{-2}	1.25
3.91 _{–4}	3.32×10^{-2}	–0.50	2.86×10^0	–0.50	5.94×10^{-2}	–0.39	1.24×10^{-1}	–0.32
1.95 _{–4}	1.29×10^{-2}	1.36	1.08×10^0	1.41	2.16×10^{-2}	1.46	3.69×10^{-2}	1.75
9.77 _{–5}	1.73×10^{-2}	–0.42	1.48×10^0	–0.46	2.94×10^{-2}	–0.45	6.11×10^{-2}	–0.73
4.88 _{–5}	8.43×10^{-3}	1.03	7.22×10^{-1}	1.04	1.42×10^{-2}	1.06	2.64×10^{-2}	1.21
2.44 _{–5}	1.02×10^{-2}	–0.27	8.72×10^{-1}	–0.27	1.71×10^{-2}	–0.27	3.28×10^{-2}	–0.31

In Table 3 we see the error in the state and control with respect to the L^2 - and L^∞ -norm and the corresponding rates of convergence on regular sparse grids with the gradient approach and fold out basis functions. The rate of convergence shows a similar behaviour in all three variables with values from 0.5 to values around 1.

For adaptive and regular sparse grids using the gradient approach and normal basis functions we observe a similar behaviour, see Fig. 4. When using fold out hat functions in comparison to normal ones we observe a slight reduction of the convergence rate for the feedback law by comparison and the amplitude of the oscillations increases when using the feedback law based on the gradient on adaptive sparse grids, see Fig. 5 and Table 4. In Fig. 6 the behaviour on regular and adaptive sparse grids with respect to the number of nodes is shown. We again observe an oscillatory behaviour when using fold out basis functions on

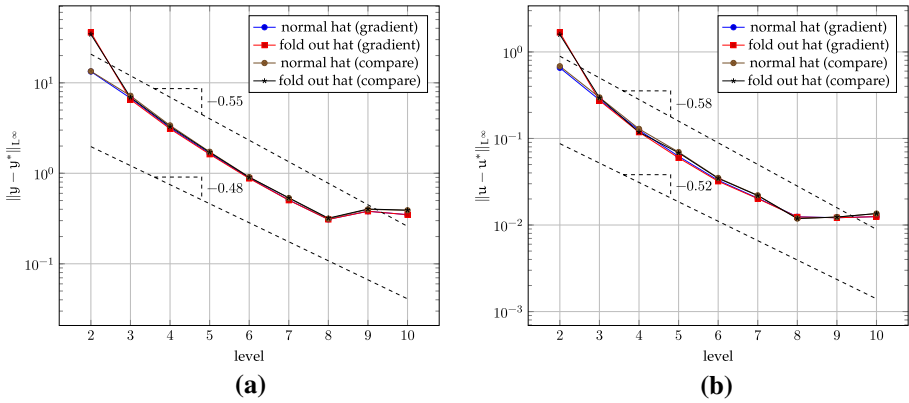


Fig. 4 Harmonic oscillator (6.9) in 2D: comparing the scheme on regular sparse grids using normal hat and fold out basis functions. **a** Error in the trajectory versus level, **b** error in the control versus level

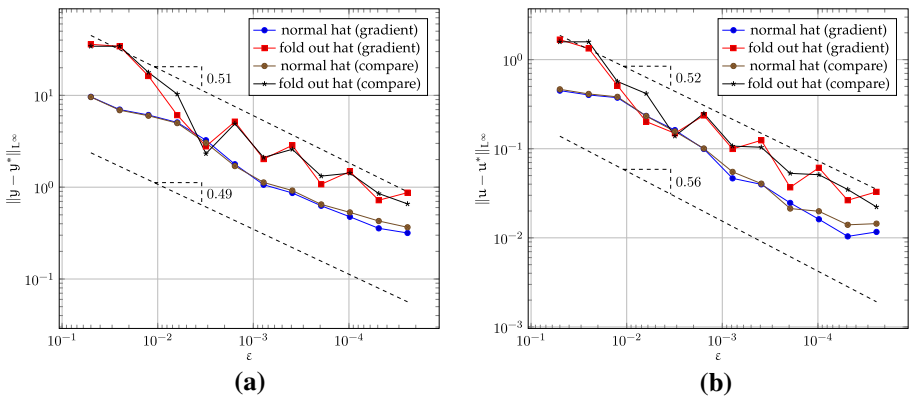


Fig. 5 Harmonic oscillator (6.9) in 2D: comparing the scheme on adaptive sparse grids using normal hat and fold out basis functions. **a** Error in the trajectory versus ϵ , **b** error in the control versus ϵ

adaptive sparse grids. The corresponding adaptive sparse mesh at initial time with normal and fold out basis functions is shown in Fig. 7.

Since we have numerically confirmed that the compare and gradient approaches behave essentially the same, we only investigate the gradient approach for the following examples to avoid the more costly comparison approach in the higher dimensional control spaces.

6.3 Control of the Semi-discrete Wave Equation (4D)

In this example we consider the optimal control problem (3.24) and (3.25) for the semi-discrete wave equation (3.10) in dimension four, that means we discretize the state and velocity by two modes using (3.7). We choose the parameters as

$$Q = [-3, 3]^4, \quad \beta_y = 2, \quad \beta_u = 0.1, \quad T = 4, \quad \Delta t = 0.01, \quad c = 0.05. \quad (6.10)$$

Since a full grid approach is very costly in four dimensions, we consider directly the error in the state and control along an optimal trajectory for a given initial point to study the convergence. For the initial point $x = (0.4, 0.6, 0, 0)^T$ we see in Tables 5 and 6 the behaviour

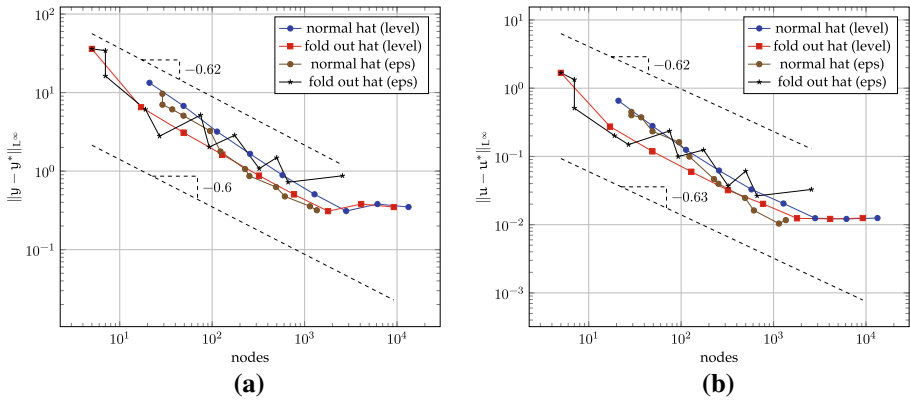


Fig. 6 Harmonic oscillator (6.9) in 2D: comparing the scheme using the gradient approach 5.2.2 on adaptive and regular sparse grids using normal hat and fold out basis functions. **a** Error in the trajectory versus nodes, **b** error in the control versus nodes

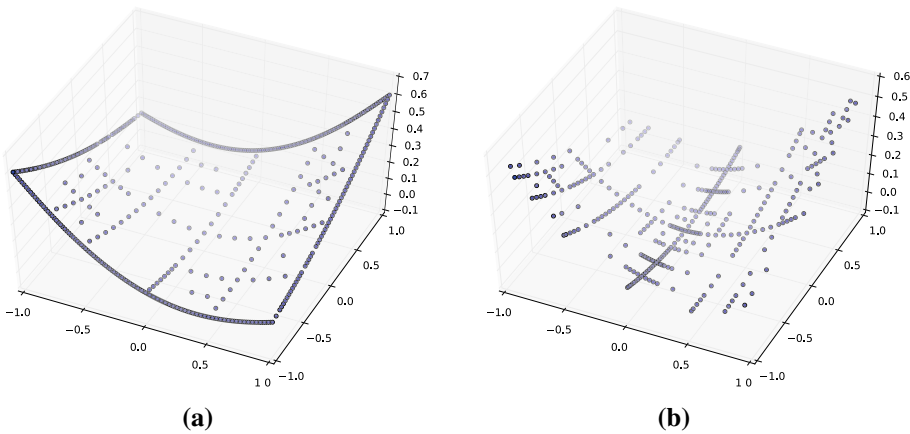


Fig. 7 Harmonic oscillator (6.9) in 2D: adaptive sparse grid for $\varepsilon = 1.95 \times 10^{-4}$. **a** Normal hat functions, **b** fold-out hat functions

of the error for $d = 2$ and $m = 2$ using fold out basis functions on regular and adaptive sparse grids. Note that when using normal basis functions on an adaptive sparse grid we observe a similar performance. In Fig. 8a we see the error on adaptive sparse grids. As before the error with respect to the threshold ε behaves similarly for normal and fold out basis function, here not only in the convergence rate, but also in absolute values. In this four dimensional example the fold out basis functions use less basis functions for the same ε , therefore when considering the error with respect to the number of nodes we observe in Fig. 8b that the fold out basis functions perform better than the normal ones. Furthermore, when comparing adaptive against regular sparse grids, we see from the figure and Tables 5 and 6 that the adaptive scheme needs an order of magnitude less points for the same accuracy. Since the absolute value of the entries in the stiffness matrix increase in higher dimension and require a finer discretization, the behaviour of the error can therefore be interpreted in the way that the adaptive algorithm automatically leads to adaptivity with respect to the different dimensions leading to anisotropic sparse grids.

Table 5 Wave equation (6.10) in 4D: error in the trajectory and control with regular sparse grids using the gradient approach and fold out basis functions

Level	e_{∞}^y	$\rho_{e_{\infty}^y}$	e_{∞}^u	$\rho_{e_{\infty}^u}$	Nodes (end)
2	6.93×10^{-1}	–	2.87×10^0	–	9
3	6.27×10^{-1}	0.14	1.79×10^0	0.68	49
4	4.40×10^{-1}	0.51	1.36×10^0	0.40	209
5	2.61×10^{-1}	0.75	5.72×10^{-1}	1.24	769
6	1.45×10^{-1}	0.85	2.95×10^{-1}	0.96	2561
7	9.00×10^{-2}	0.68	1.95×10^{-1}	0.59	7937
8	4.53×10^{-2}	0.99	9.47×10^{-2}	1.05	23,297
9	1.42×10^{-2}	1.68	3.28×10^{-2}	1.53	65,537
10	6.01×10^{-3}	1.24	1.43×10^{-2}	1.19	1.78×10^5

Table 6 Wave equation (6.10) in 4D: error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions

ε	e_{∞}^y	$\rho_{e_{\infty}^y}$	e_{∞}^u	$\rho_{e_{\infty}^u}$	Nodes (end)
5.00 ₋₂	3.67×10^{-1}	–	3.05×10^0	–	389
2.50 ₋₂	2.52×10^{-1}	0.54	1.28×10^0	1.25	567
1.25 ₋₂	1.38×10^{-1}	0.87	1.07×10^0	0.27	655
6.25 ₋₃	1.35×10^{-1}	0.03	3.46×10^{-1}	1.62	889
3.13 ₋₃	1.39×10^{-1}	–0.04	3.29×10^{-1}	0.07	1051
1.56 ₋₃	1.41×10^{-1}	–0.02	4.50×10^{-1}	–0.45	1231
7.81 ₋₄	8.73×10^{-2}	0.70	1.42×10^{-1}	1.67	1671
3.91 ₋₄	6.33×10^{-2}	0.46	7.37×10^{-2}	0.94	2207
1.95 ₋₄	4.53×10^{-2}	0.48	5.23×10^{-2}	0.49	3581
9.77 ₋₅	3.09×10^{-2}	0.55	3.59×10^{-2}	0.54	5133
4.88 ₋₅	1.94×10^{-2}	0.67	3.04×10^{-2}	0.24	9829
2.44 ₋₅	1.27×10^{-2}	0.61	1.84×10^{-2}	0.73	13,249

In Fig. 9 we confirm that the feedback control we derive from the value function leads to a trajectory which coincide (up to a small error depending on the mesh parameter) with a reference trajectory derived from a Riccati equation.

6.4 Control of the Semi-discrete Wave Equation (6D)

Next, we consider problem (3.24) and (3.25) for the semi-discrete wave equation (3.10) in six dimensions with data as given in (6.10) and initial point $x = (0.4, 0.6, 0.2, 0, 0, 0)^T$. Note that we have to decrease the size of the time step to $\Delta t = 0.0025$ when increasing the entries in the stiffness matrix. We remark that larger entries in the stiffness matrix lead to an increase of the derivative $|H_p(x, p)|$, which requires in the context of finite difference schemes a smaller time step because of the CFL-condition. As before we use the gradient approach to derive the feedback and now employ only fold out basis function to keep the

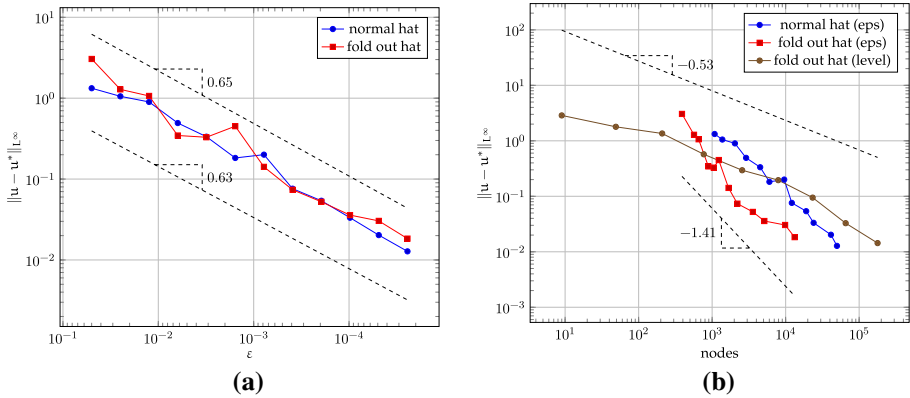


Fig. 8 Wave equation (6.10) in 4D: error in the control comparing the scheme based on sparse grids using normal hat and fold out basis functions, regular and adaptive sparse grids. **a** Error in the control versus ϵ , **b** error in the control versus nodes

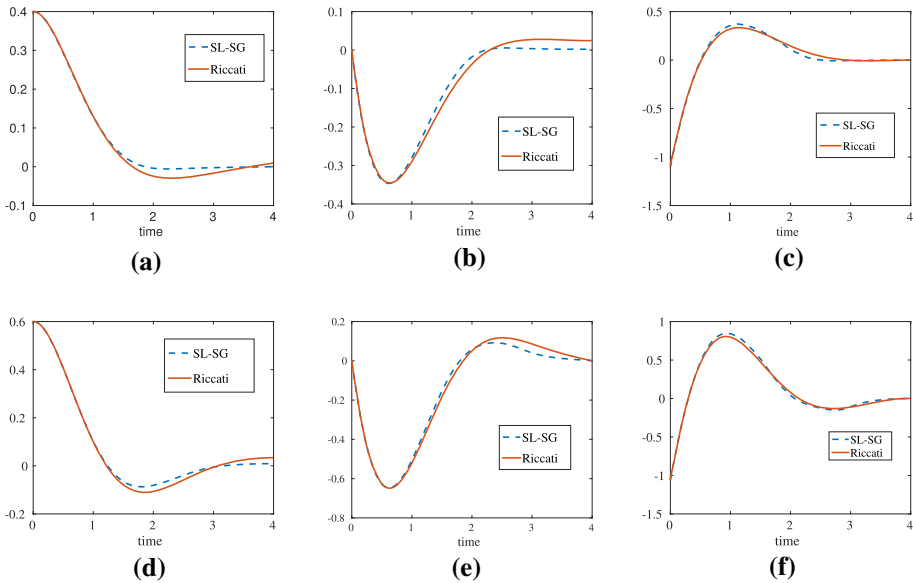


Fig. 9 Wave equation (6.10) in 4D: Components of the optimal state y and corresponding optimal control u resulting from the SL-SG approach on a sparse grid with threshold $\epsilon = 2.44 \times 10^{-5}$. As a reference solution the corresponding components of a solution generated by a Riccati approach are presented. **a** y_1 , **b** y_3 , **c** u_1 , **d** y_2 , **e** y_4 , **f** u_2

computational cost low. In Table 7 and Fig. 10 the behaviour in the state along an optimal trajectory is shown, overall similar to the 4D case, although a certain resolution needs to be there for the convergence to kick in.

Table 7 Wave equation (6.10) in 6D: error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions

ε	e_∞^y	$\rho_{e_\infty^y}$	e_∞^u	$\rho_{e_\infty^u}$	Nodes (end)
5.00 ₋₂	5.75×10^{-1}	–	1.69×10^0	–	43
2.50 ₋₂	5.78×10^{-1}	–0.01	1.70×10^0	–0.01	69
1.25 ₋₂	1.51×10^{-1}	1.94	2.33×10^0	–0.45	1015
6.25 ₋₃	1.18×10^{-1}	0.36	1.77×10^0	0.40	919
3.13 ₋₃	1.11×10^{-1}	0.09	3.07×10^{-1}	2.52	1549
1.56 ₋₃	1.14×10^{-1}	–0.04	4.15×10^{-1}	–0.44	1779
7.81 ₋₄	1.06×10^{-1}	0.10	3.13×10^{-1}	0.41	2053
3.91 ₋₄	7.73×10^{-2}	0.46	2.12×10^{-1}	0.56	2851
1.95 ₋₄	4.80×10^{-2}	0.69	1.03×10^{-1}	1.04	3657
9.77 ₋₅	2.98×10^{-2}	0.69	5.35×10^{-2}	0.95	5415
4.88 ₋₅	2.35×10^{-2}	0.34	3.68×10^{-2}	0.54	9691
2.44 ₋₅	1.59×10^{-2}	0.57	2.59×10^{-2}	0.51	12,507

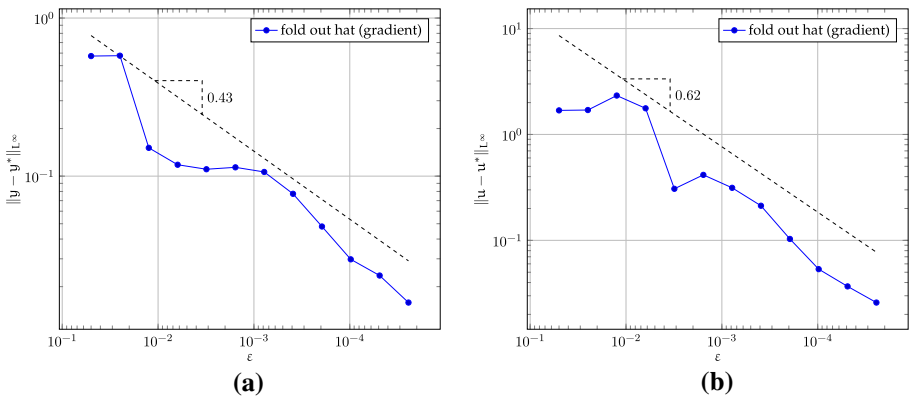


Fig. 10 Wave equation (6.10) in 6D: the adaptive sparse grids scheme using fold out basis functions. **a** error in the trajectory versus ε , **b** error in the control versus ε

6.5 Control of the Semi-discrete Wave Equation (8D)

Next, we consider problem (3.24) and (3.25) for the semi-discrete wave equation (3.10) in eight dimension with data as given in (6.10) and initial point $x = (0.4, 0.6, 0.2, 0.1, 0, 0, 0, 0)^T$. Again, we needed to decrease the time step, now to $\Delta t = 0.00125$. In Fig. 11 and Table 8 we see the errors with respect to the threshold ε . Again, the approach needs a certain resolution for a stable convergence behaviour, which now starts for a smaller ε than in the 6D case. The number of mesh points on the finest refinement level is still moderate, however the computational cost is relatively high.

This being our example with the largest number of dimensions, we note that we needed about 2.5 GB of main memory during the computation for the finest resolution.

Table 8 Wave equation (6.10) in 8D: error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions

ε	e_{∞}^y	$\rho_{e_{\infty}^y}$	e_{∞}^u	$\rho_{e_{\infty}^u}$	Nodes (end)
5.00 ₋₂	5.64×10^{-1}	–	1.63×10^0	–	49
2.50 ₋₂	5.74×10^{-1}	–0.03	1.68×10^0	–0.04	63
1.25 ₋₂	5.77×10^{-1}	–0.01	3.32×10^0	–0.98	241
6.25 ₋₃	1.40×10^{-1}	2.04	2.51×10^0	0.40	1391
3.13 ₋₃	1.17×10^{-1}	0.26	1.78×10^0	0.50	1455
1.56 ₋₃	1.15×10^{-1}	0.03	3.57×10^{-1}	2.32	2169
7.81 ₋₄	1.08×10^{-1}	0.09	3.56×10^{-1}	0.00	2759
3.91 ₋₄	1.01×10^{-1}	0.11	3.81×10^{-1}	–0.10	3283
1.95 ₋₄	7.26×10^{-2}	0.47	2.18×10^{-1}	0.81	5011
9.77 ₋₅	3.89×10^{-2}	0.90	6.92×10^{-2}	1.65	7113
4.88 ₋₅	2.90×10^{-2}	0.43	4.98×10^{-2}	0.47	9059

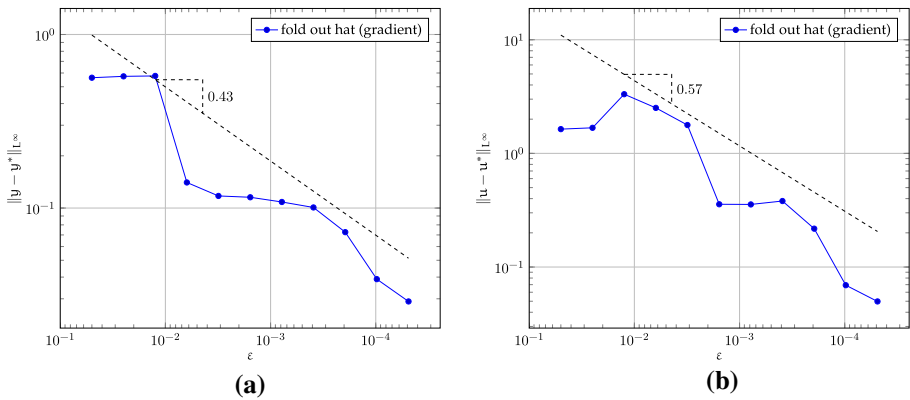


Fig. 11 Wave equation (6.10) in 8D: the adaptive sparse grids scheme using fold out basis functions. **a** Error in the trajectory versus ε , **b** error in the control versus ε

6.6 Control of a Bilinear System (2D)

In this last example we consider a dynamics arising from a bilinear dynamical system which is of Schrödinger type, see (3.19). Because of the nonlinear coupling of state and control the Riccati approach is here not directly applicable.

We consider two settings; while in the first one we observe numerically a smooth value function, in the second one a non-differentiability appears.

Remark 1 If the value function is not differentiable in a point (x, t) , then (x, t) is not a regular point, i.e. there exists not a unique optimal trajectory $y^*(\cdot)$ starting in x at time t with $y^*(t) = x$, see [18, p. 42, Thm. 10.2].

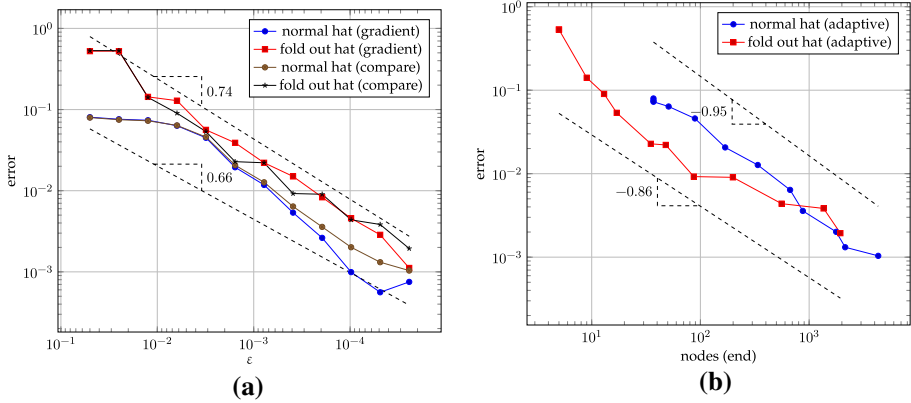


Fig. 12 Bilinear equation (6.11): error in the value function comparing the scheme based on sparse grids using normal and fold out basis functions, adaptive sparse grid (2D). **a** Error in the value function versus ϵ , **b** error in the value function versus nodes

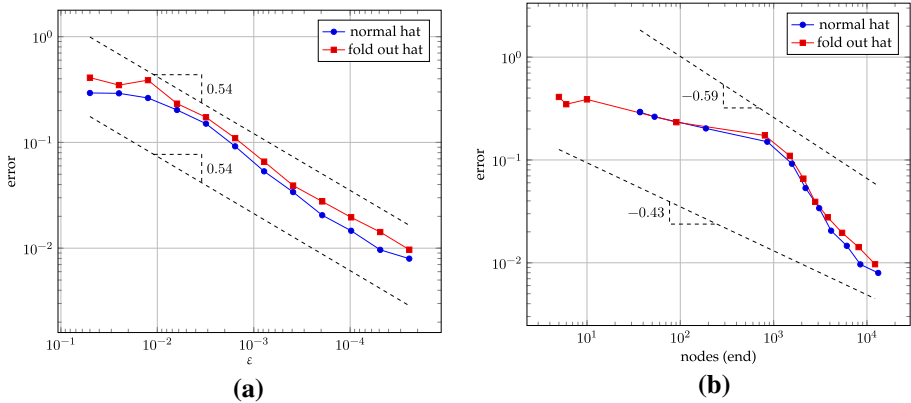


Fig. 13 Bilinear equation (6.12) in 2D: error in the value function comparing the scheme based on sparse grids using normal and fold out basis functions, adaptive sparse grid. **a** Error in the value function versus ϵ , **b** error in the value function versus nodes

Example 1 Let $Q = [-2, 2]^2$, $T = 1$, $\Delta t = 1/200$, $\beta_y = 2$, $\beta_u = 0.1$,

$$A^s = \begin{pmatrix} 0 & 0.1 \\ -0.1 & 0 \end{pmatrix}, \quad \hat{B} = \begin{pmatrix} 0 & -0.1 \\ 0.1 & 0 \end{pmatrix}, \quad \mathcal{M} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad (6.11)$$

and control bounds inactive, i.e. we choose $U_s = [-10, 10]$ big enough for the comparison approach. For computing a reference value function as described in Sect. 6.1 we choose Δx and Δt as given in Sect. 6.2. In Fig. 12 we see the error when using comparison and the gradient. Using normal/fold out basis functions and a comparison/gradient based approach lead to a similar behaviour of the error in the value function with respect to the threshold ϵ .

Example 2 Let

$$A^s = \begin{pmatrix} 0 & 0.5 \\ -0.5 & 0 \end{pmatrix}, \quad \hat{B} = \begin{pmatrix} 0 & -0.5 \\ 0.5 & 0 \end{pmatrix}, \quad (6.12)$$

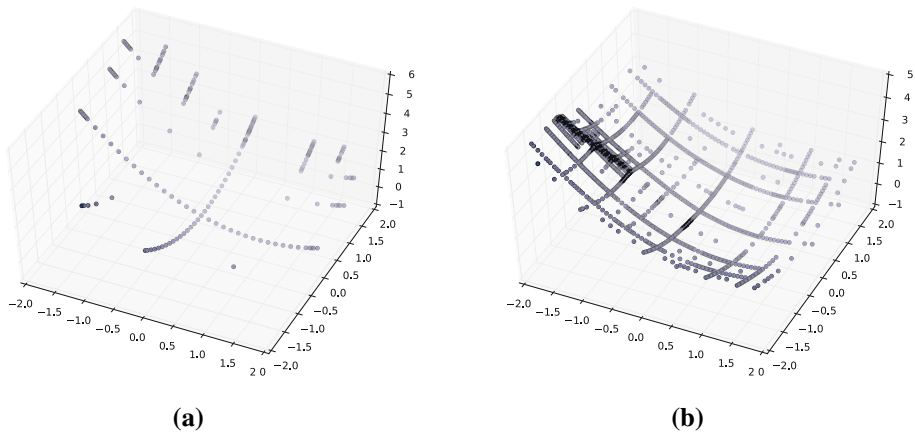


Fig. 14 Bilinear equation (6.11) and (6.12) in 2D: adaptive sparse grid. **a** Example (6.11) with feedback (5.3), fold-out basis functions and $\varepsilon = 1.95 \times 10^{-4}$, **b** example (6.12) with feedback (5.3), fold-out basis functions and $\varepsilon = 7.78 \times 10^{-4}$

and, besides $\Delta t = 1/500$, the other parameters as in Example 1, and let the control space be given by $U = [-4, 4]$. In Fig. 13 we see the behaviour of the error when using the comparison approach and normal and fold out basis functions, respectively. In Fig. 14 the meshes for $\varepsilon = 1.95 \times 10^{-4}$ and $\varepsilon = 7.78 \times 10^{-4}$ are shown. We observe a strong refinement of the mesh along the non-differentiability.

In both example we observe a similar convergence behaviour as for the wave equation. Furthermore, in this two dimensional setting again the different approaches lead to quite the same results.

7 Outlook

The presented approach allows to include further aspects to increase the accuracy and to reduce the computational costs. A higher order interpolation could be used within the semi-Lagrangian scheme (cf. also [46]). Additionally an efficient numerical algorithm to determine the minimum within the Hamiltonian in the semi-Lagrangian scheme could be developed which is also suitable for a more general class of cost functionals. Finally information on the basis function could be used to introduce a priori an adaptivity with respect to the dimensions.

Note that the computational bottleneck is the evaluation of the sparse grid function in the scheme (5.1). Our current proof-of-concept implementation is essentially unoptimized for this aspect. The more evolved and sophisticated SG++-library [43] already would most likely show a much reduced computational time. Furthermore, there are recent results for more efficient and suitable data structures for sparse grids which take machine constraints into consideration [12], which have the potential to gain a factor of two to three in serial runtime against the mature SG++, while allowing for multicore parallelism. In conclusion, although the runtime for the eight dimensional problem is in our implementation in the order of days, it is fully justified to assume it can be reduced to hours, if not less, on multicore architectures.

Besides an efficient implementation of the sparse grid function evaluation, two other aspects warrant further investigation. One is the interplay between spatial and time resolution.

While in this work we concentrated on the spatial resolution, it is well known that these discretizations should be analyzed jointly. Here, the non-local nature of the sparse grid basis function pose additional difficulties in the analysis. The other is the spatial adaptivity, here we concentrated on using the hierarchical coefficients, but these refine in all dimensions, which results in the examples in overrefinement in some dimensions due to anisotropic nature of the problem. Only through the coarsening these grid points will be removed again, but of course it would be much more efficient, if the adaptive procedure only refines in some dimensions.

References

1. Alla, A., Falcone, M.: An adaptive POD approximation method for the control of advection-diffusion equations. *Int. Ser. Numer. Math.* **164**, 1–17 (2013)
2. Alla, A., Falcone, M., Kalise, D.: HJB-POD based feedback design approach for the wave equation. *Bull. Braz. Math. Soc.* **47**(1), 51–64 (2016)
3. Aronna, M.S., Bonnans, J.F., Kröner, A.: Optimal Control of PDEs in a Complex Space Setting; Application to the Schrödinger Equation. Research report, INRIA, 2016, HAL report: hal-01311421. https://hal.archives-ouvertes.fr/hal-01311421/file/ABK_2016_v1.pdf
4. Bardī, M., Capuzzo-Dolcetta, I.: Optimal Control and Viscosity Solutions of Hamilton–Jacobi–Bellman Equations. Birkhäuser, Boston (2008)
5. Barles, G., Souganidis, P.E.: Convergence of approximation schemes for fully nonlinear second order equations. *Asymptot. Anal.* **4**(3), 271–283 (1991)
6. Baur, U., Benner, P.: Modellreduktion für parametrisierte Systeme durch balanciertes Abschneiden und Interpolation (Model reduction for parametric systems using balanced truncation and interpolation). *at-Automatisierungstechnik* **57**8, 411–420 (2009)
7. Beauchard, K., Nersisyan, V.: Semi-global weak stabilization of bilinear Schrödinger equations. *C. R. Math.* **348**(19–20), 1073–1078 (2010)
8. Bokanowski, O., Desilles, A., Zidani, H.: ROC-HJ-Solver. A C++ Library for Solving HJ Equations (2013). <http://uma.ensta-paristech.fr/soft/ROC-HJ/>
9. Bokanowski, O., Forcadell, N., Zidani, H.: Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM J. Control Optim.* **48**(7), 4292–4316 (2010)
10. Bokanowski, O., Garcke, J., Griebel, M., Klompaker, I.: An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton–Jacobi Bellman equations. *J. Sci. Comput.* **55**(3), 575–605 (2013)
11. Bungartz, H.-J., Griebel, M.: Sparse grids. *Acta Numer.* **13**, 1–123 (2004)
12. Buse, G.: Exploiting Many-Core Architectures for Dimensionally Adaptive Sparse Grids. Dissertation, Institut für Informatik, Technische Universität München, München (2015)
13. Carlini, E., Falcone, M., Ferretti, R.: An efficient algorithm for Hamilton–Jacobi equations in high dimension. *Comput. Vis. Sci.* **7**(1), 15–29 (2004)
14. Chkifa, A., Cohen, A., Schwab, C.: High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Found. Comput. Math.* **14**(4), 601–633 (2014)
15. Falcone, M., Ferretti, R.: Semi-Lagrangian approximation schemes for linear and Hamilton–Jacobi equations. In: *Applied Mathematics. Society for Industrial and Applied Mathematics*, Philadelphia (2013)
16. Ferretti, R.: Internal approximation schemes for optimal control problems in Hilbert spaces. *J. Math. Syst. Estim. Control* **7**(1), 1–25 (1997)
17. Feuersänger, C.: Sparse Grid Methods for Higher Dimensional Approximation. Dissertation, Institut für Numerische Simulation, Universität Bonn (2010)
18. Fleming, W.H., Soner, H.M.: Controlled Markov processes and viscosity solutions. In: *Stochastic Modelling and Applied Probability*. Springer, New York (2006)
19. Garcke, J.: Sparse grids in a nutshell. In: Garcke, J., Griebel, M. (eds.) *Sparse Grids and Applications. Lecture Notes in Computational Science and Engineering*, vol. 88, pp. 57–80. Springer, Berlin (2012)
20. Garcke, J., Griebel, M.: On the computation of the eigenproblems of hydrogen and helium in strong magnetic and electric fields with the sparse grid combination technique. *J. Comput. Phys.* **165**(2), 694–716 (2000)
21. Gibson, J.S.: An analysis of optimal modal regulation: convergence and stability. *SIAM J. Control Optim.* **19**(5), 686–707 (1981)
22. Gombao, S.: Approximation of optimal controls for semilinear parabolic PDE by solving Hamilton–Jacobi–Bellman equations. In: *Proceedings of the 15th International Symposium on the Mathematical Theory of Networks and Systems*, of Notre Dame, South Bend, IN, USA (2002)

23. Griebel, M.: A parallelizable and vectorizable multi-level algorithm on sparse grids. In: Hackbusch, W. (ed.) *Parallel Algorithms for Partial Differential Equations*, Notes on Numerical Fluid Mechanics, vol. 31, pp. 94–100. Vieweg, Braunschweig (1991)
24. Griebel, M., Hamaekers, J.: Sparse grids for the Schrödinger equation. *Math. Model. Numer. Anal.* **41**(2), 215–247 (2007)
25. Griebel, M., Knapek, S.: Optimized general sparse grid approximation spaces for operator equations. *Math. Comput.* **78**(268), 2223–2257 (2009)
26. Hinze, M., Volkwein, S.: Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control. In: Benner, P., Mehrmann, V., Sorensen, D.C. (eds.) *Dimension Reduction of Large-Scale Systems*. Lecture Notes in Computer Science Engineering, vol. 45, pp. 261–306. Springer, Berlin (2005)
27. Horowitz, M.B., Damle, A., Burdick, J.W.: Linear Hamilton Jacobi Bellman equations in high dimensions. In: *IEEE 53rd Annual Conference on Decision and Control (CDC)*, pp. 5880–5887 (2014)
28. Hu, C., Shu, C.: A discontinuous Galerkin finite element method for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* **21**(2), 666–690 (1999)
29. Ishii, H.: Uniqueness of unbounded viscosity solution of Hamilton–Jacobi equations. *Indiana Univ. Math. J.* **33**(5), 721–748 (1984)
30. Kalise, D., Kröner, A., Kunisch, K.: Local minimization in dynamic programming equations. *SIAM J. Sci. Comput.* (2016) (**preprint**). <https://www.ricam.oeaw.ac.at/publications/reports/15/rep15-04.pdf>
31. Kröner, A., Kunisch, K., Zidani, H.: Optimal feedback control of the undamped wave equation by solving a HJB equation. *ESAIM Control Optim. Calc. Var.* **21**(2), 442–464 (2015)
32. Kunisch, K., Volkwein, S., Xie, L.: HJB-POD based feedback design for the optimal control of evolution problems. *SIAM J. Appl. Dyn. Syst.* **4**, 701–722 (2004)
33. Kunisch, K., Xie, L.: POD-based feedback control of the Burgers equation by solving the evolutionary HJB equation. *Comput. Math. Appl.* **49**, 1113–1126 (2005)
34. Lasiecka, I., Triggiani, R.: *Control of Partial Differential Equations: Continuous and Approximation Theories*, vol. I. Cambridge University Press, Cambridge (2000)
35. Lions, J.-L., Magenes, E.: *Non-homogeneous Boundary Value Problems and Applications*, vol. I. Springer, Berlin (1972)
36. Lions, J.L.: *Optimal Control of Systems Governed by Partial Differential Equations*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin (1971)
37. Mirrahimi, M., Handel, R.V.: Stabilizing feedback controls for quantum systems. *SIAM J. Control Optim.* **46**(2), 445–467 (2007)
38. Mirrahimi, M., Rouchon, P., Turinici, G.: Lyapunov control of bilinear Schrödinger equations. *Automatica* **41**(11), 1987–1994 (2005)
39. Morris, K.A.: Control of systems governed by partial differential equations. In: Levine, W.S. (ed.) *The IEEE Control Theory Handbook*. CRC Press, Boca Raton (2010)
40. Novak, E., Ritter, K.: Global optimization using hyperbolic cross points. In: Floudas, C.A. (eds.) *State of the Art in Global Optimization: Computational Methods and Applications*. Nonconvex Optim. Appl., vol. 7, pp. 19–33. Kluwer, Dordrecht (1996)
41. Osher, S., Shu, C.W.: High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **28**(4), 907–922 (1991)
42. Peherstorfer, B., Zimmer, S., Bungartz, H.-J.: Model reduction with the reduced basis method and sparse grids. In: Garcke, J., Griebel, M. (eds.) *Sparse Grids and Applications*. LNCSE, vol. 88, pp. 223–242. Springer, Berlin (2013) (**English**)
43. Pflüger, D.: *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Dissertation, Institut für Informatik, Technische Universität München, München (2010)
44. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk. SSSR* **148**, 1042–1043 (1963) (**Russian**), Engl. Transl.: *Soviet Math. Dokl.* 4:240–243 (1963)
45. Springer, R.: *Lösung von Hamilton–Jacobi–Bellman–Gleichungen auf dünnen Gittern*. Diplomarbeit, University Chemnitz (2013)
46. Warin, X.: *Adaptive Sparse Grids for Time Dependent Hamilton–Jacobi–Bellman Equations in Stochastic Control* (2014). [arXiv:1408.4267](https://arxiv.org/abs/1408.4267)
47. Zenger, C.: Sparse grids. In: Hackbusch, W. (ed.) *Parallel Algorithms for Partial Differential Equations*. Notes on Numerical Fluid Mechanics, vol. 31, Vieweg, Braunschweig, pp. 241–251 (1991)