

Suboptimal feedback control of PDEs by solving HJB equations on adaptive sparse grids

Jochen Garcke, Axel Kröner

▶ **To cite this version:**

Jochen Garcke, Axel Kröner. Suboptimal feedback control of PDEs by solving HJB equations on adaptive sparse grids. Journal of Scientific Computing, 2017, 70 (1), pp.1-28. hal-01185912

HAL Id: hal-01185912
https://hal.science/hal-01185912

Submitted on 7 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

18 August, 2015

SUBOPTIMAL FEEDBACK CONTROL OF PDES BY SOLVING HJB EQUATIONS ON ADAPTIVE SPARSE GRIDS

JOCHEN GARCKE AND AXEL KRÖNER

Abstract. An approach to solve finite time horizon sub-optimal feedback control problems for partial differential equations is proposed by solving dynamic programming equations on adaptive sparse grids. The approach is illustrated for the wave equation and an extension to equations of Schrödinger type is indicated. 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. 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.

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

$$\begin{cases} \min\limits_{u\in U_{\text{ad}}} J(u) = \displaystyle\int_0^T l(y(t),u(t))\mathrm{d}t, \\ y_t(t) = f(y(t),u(t)), \\ y(0) = y_0, \end{cases} \tag{1.1}$$

with dynamics $f\colon \mathbb{R}^d\times\mathbb{R}^m \to \mathbb{R}$, $d,m\in\mathbb{N}$, which arise from the semi-discretization of a PDE, running cost $l\colon \mathbb{R}^d\times\mathbb{R}^m\to\mathbb{R}$, 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 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

Key words and phrases. sub-optimal feedback control of PDEs, Hamilton-Jacobi Bellman equation, sparse grids, curse of dimensionality.

be considered to control the solution of the corresponding semi-discrete systems – without using Hamilton-Jacobi Bellman equations – the presented approach aims to give a general framework which is applicable on a wider class of problems.

To set up a feedback law we characterize the value function of problem (1.1) as the viscosity solution of an instationary Hamilton-Jacobi Bellman (HJB) equation, from which we derive a control in feedback form. However, to solve the equation numerically we have to solve a non-linear partial differential equation, which is by the curse of dimensionality very challenging in higher dimension. To reduce the dimensionality of the problem there exists 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., [39, 41]. Furthermore one can use discretizations of the PDE which represent the dynamics of interest (e.g. the lower modes) and use a discretization based on spectral elements, cf. [37]. 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 [47], semi-Lagrangian [13, 15], discontinuous Galerkin methods [30], sparse grids [9], or low rank tensor approximation for linear HJB equations [29].

In this paper we use spectral elements for the discretization of the underlying PDE following [37] (i.e. the aim is to control the lower modes) and we solve the corresponding HJB equation by an adaptive semi-Lagrangian scheme on sparse grids based on [9]. 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 a sufficiently smooth function f 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 $O(h_n^2(\log h_n^{-1})^{d-1})$. The approach requires H^2_{mix} regularity of the value function. Sparse grids go back to the work by Smolyak [49] and in particular Zenger [52], Griebel [24], 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, 48] for references on adaptive sparse grids. Note that since the interpolation on sparse grids is not monotone [9, 48] the scheme is not monotone, and a convergence theorem follows not directly from Barles & Souganidis [4].

In this paper we consider control problems for the wave and the bilinear Schrödinger equation. In contrast to [9] 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 complexity in higher dimensions, 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 about some references. For optimal feedback control problems by solving HJB equations for reduced systems using proper orthogonal decomposition we refer to [1, 2, 23, 28, 39, 41], for spectral elements to [37], and for balanced truncation to [33]. 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, 20, 24, 26, 48, 52]. For sparse approximation of PDEs in high dimensions, see, e.g., [14] and for sparse grids methods for solving Schrödinger equations we refer to [21, 25]. Higher order semi-Lagrangian schemes on sparse grids for second order HJB equations are considered in [51]. Furthermore, in [50] several two dimensional numerical examples for semi-Lagrangian schemes on sparse grids for first order HJB equation are considered. For optimal control of the wave equation, see, e.g., [22, 27, 35, 36, 38, 40] and for optimal control of the Schrödinger equation, see, e.g., [5, 6, 10, 19, 32, 44, 45].

The paper is organized as follows. In Section 3 we formulate the closed-loop optimal control problems arising from control of the wave and Schrödinger equation, in Section 4 we recall the basic ideas of sparse grids, in Section 5 we formulate the semi-Lagrangian scheme, and in Section 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 $H^m(\Omega) = H^m(\Omega, \mathbb{R})$ (resp. $\bar{H}^m(\Omega) = H^m(\Omega, \mathbb{C})$), $m \in \mathbb{N}$. We set $H = L^2(\Omega, \mathbb{R})$ and $\bar{H} = L^2(\Omega, \mathbb{C})$. Furthermore, we introduce for 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 $\|\cdot\|_\infty$. For a set $\Omega = [a, b]^d$ with $a, b \in \mathbb{R}^d$ and $d \in \mathbb{N}$, we define $\text{length}(\Omega) = \max_{i=1,\dots,d} |a_i - b_i|$.

3. Optimal control problem and its approximation

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

3.1. Optimal control problem.

Let $0 < T < \infty$, $L > 0$, $c > 0$. We define the set of admissible controls $U_{\text{ad}} = L^2(0, T; U^m)$ with

$$U^m := \{u \in \mathbb{R}^m |\ u_a \le u \le u_b\} \tag{3.1}$$

for $u_a, u_b \in \mathbb{R}^m$ and $m \in \mathbb{N}$. Further we set $U_w := U^m$, and $U_s := U^1$. For controls $u \in L^2(0, T; U_w)$ and $\Omega \subset (0, T)$ we formulate the wave equation given by

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

for initial state and velocity $\hat{y}_0 \in H^1(\Omega)$ and $\hat{y}^1 \in L^2(\Omega)$, and control operator

$$B := (\sin(\pi x), \dots, \sin(m\pi y)). \tag{3.3}$$

We remark that the presented approach can be easily extended to more general control operators B. Equation (3.2) has a unique solution in $C(0, T; H^1(\Omega)) \cap C^1(0, T; L^2(\Omega))$, see [42, pp. 275, 288].

For controls $u \in L^2(0,T;U_s)$ and $\Omega \subset \mathcal{C}$ bounded domain we formulate the Schrödinger equation given by

$$\begin{cases} i\hat{y}_t + c\Delta\hat{y} - u\hat{B}\hat{y} = 0 & \text{in } (0,T)\times\Omega, \\ \hat{y}(0) = \hat{y}_0 & \text{in } \Omega, \\ \hat{y} = 0 & \text{on } (0,T)\times\partial\Omega, \end{cases} \tag{3.4}$$

for initial state $\hat{y}_0 \in \bar{H}^1(\Omega)$, and control operator $B \in \mathcal{L}(L^2(\Omega))$. Equation (3.4) has a unique solution in $C(0,T;\bar{H}^1(\Omega)) \cap C^1(0,T;\bar{H}^{-1}(\Omega))$, see [42]. There holds the property that $\|y(t)\|_{\bar{H}} = \|y_0\|_{\bar{H}}$ for all $t \in [0,T]$.

Let the cost functional be given by

$$J(u,\hat{y}) := \int_0^T \left(\alpha_1 \|\hat{y}(t) - y_\mathrm{d}\|_{\mathcal{H}}^2 + \alpha \|u(t)\|_U^2\right) \mathrm{d}t \tag{3.5}$$

with $\mathcal{H} = H$, $U = U_w$ for the wave equation (resp. $\mathcal{H} = \bar{H}$, $U = U_s$ for the Schrödinger equation), $y_\mathrm{d} \in H$ (resp. $y_\mathrm{d} \in \bar{H}$), as well as $\alpha_1 > 0$ and $\alpha > 0$. We denote the control-to-state operator for the wave and Schrödinger equation by $\hat{y}[\cdot]$. The optimal control problem is given by

$$\min F(u) := J(u,\hat{y}[u]), \quad u \in L^2(0,T;U). \tag{3.6}$$

For existence of a solution of the control problem (3.6) for the wave equation, see, e.g., [43]. The existence of an optimal control for the Schrödinger equation can be shown using classical arguments: for bounded sequences in the control space we can choose weakly*-converging subsequences for which the corresponding states converge strongly in $L^2(0,T;\bar{H})$ and weakly in $C(0,T;\bar{H})$. Then the existence can be deduced using the weak lower semi-continuity of the cost functional. We remark that the optimal control for the control problem with the wave equation is unique.

To derive a feedback law we consider a semi-discrete formulation of the control problem, in particular we apply the methods of lines. For a given basis

$$b := (\varphi_1, \ldots, \varphi_d), \quad d \in \mathbb{N}, \tag{3.7}$$

with $\varphi\colon \Omega \to \mathbb{R}$ we define

$$\begin{aligned} A &:= ((\nabla\varphi_i(x), \nabla\varphi_j(x))_{i,j=1,\ldots,d}) \quad \text{(stiffness matrix)}, \\ M &:= ((\varphi_i(x), \varphi_j(x))_{i,j=1,\ldots,d}) \qquad\quad \text{(mass matrix)}. \end{aligned} \tag{3.8}$$

In the numerical examples we will choose

$$\varphi_i(x) := \sin(i\pi x), \quad i = 1, \ldots, d, \tag{3.9}$$

and obtain, cf. [37],

$$A = \mathrm{diag}((1/2(i\pi)^2)_{i=1,\ldots,d}), \quad M = \mathrm{diag}((1/2)_{i=1,\ldots,d}). \tag{3.10}$$

For the approximation of the wave equation we consider a first order system in time given by

$$\dot{y}(t) = f(y(t),u(t)), \quad t > 0, \quad y(0) = y_0 \tag{3.11}$$

with dynamics

$$f_\mathrm{w}\colon \mathbb{R}^{2d} \times \mathbb{R}^m \to \mathbb{R}, \quad f(x,u) = f_\mathrm{w}(x,u) := \mathcal{A}x + \mathcal{B}u \tag{3.12}$$

and

$$\mathcal{A} := \begin{pmatrix} 0 & I_d \\ -cM^{-1}A & 0 \end{pmatrix}, \quad \mathcal{B} := \begin{pmatrix} 0 \\ b \end{pmatrix}, \quad b \in \mathbb{R}^{m\times d}, \quad y_0 \in \mathbb{R}^{2d}. \tag{3.13}$$

For the approximation of the Schrödinger equation we write the equation as a real-valued system and consider

$$f_{\mathrm{s}} \colon \mathbb{R}^{2d} \times \mathbb{R} \to \mathbb{R}, \quad f_{\mathrm{s}}(x,u) := \mathcal{A}x + u\mathcal{B}y \tag{3.14}$$

with

$$\mathcal{A} := \begin{pmatrix} 0 & cM^{-1}A \\ -cM^{-1}A & 0 \end{pmatrix}, \quad \mathcal{B} := \mathcal{M}^{-1}\hat{\mathcal{B}}, \quad \mathcal{M} := \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}, \quad \hat{B} \in \mathbb{R}^{2d\times 2d}. \tag{3.15}$$

The discrete cost functional is given by

$$J(u,y) := \int_0^T l(y(t),u(t))\mathrm{d}t \tag{3.16}$$

with running cost $l = l_w$ (wave equation) and $l = l_s$ (Schrödinger equation)

$$l_w(x,u) := \beta_x x_1^T M x_1 + \beta_u u^T u, \quad l_s(x,u) := \beta_x x^T \mathcal{M} x + \beta_u u^T u, \tag{3.17}$$

where $x = (x_1, x_2) \in \mathbb{R}^{2d}$, $u \in U$. Making the dependence on the initial time explicit we introduce the value function

$$v(x,t) := \inf_{u\in U} \int_t^T l(y(s),u(s))\mathrm{d}s, \quad \text{s.t.} \tag{3.18}$$

$$\dot{y}(s) = f(y(s),u(s)), \quad y(t) = x, \quad s > t, \tag{3.19}$$

for $t \in [0,T]$ and $x \in \mathbb{R}^{2d}$. The value function satisfies the dynamic programming principle

$$v(x,t) = \inf_{u\in U} \left(\int_t^\tau l(y(s),u(s))\mathrm{d}s + v(y(\tau),\tau) \right) \quad \text{in } \mathbb{R}^{2d}\times[0,T], \tag{3.20}$$

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

$$\begin{cases} -v_t(x,t) + H(x,\nabla v) = 0 & \text{in } \mathbb{R}^{2d}\times[0,T], \\ v(x,T) = 0 & \text{in } \mathbb{R}^{2d}, \end{cases} \tag{3.21}$$

with Hamiltonian

$$H(x,p) := \sup_{u\in U} \left(-f(x,u)^T p - l(x,u) \right) \tag{3.22}$$

for $x \in \mathbb{R}^{2d}$ and $p \in \mathbb{R}^{2d}$, see [3, 31].

Now, to derive a feedback control law from the value function we introduce the set-valued map (following the notation in [18]) given by

$$g^*(x,t) := \mathrm{argmin}_{u\in U}\{-f(x,u)^T \nabla_x v(x,t) - l(x,u)\}. \tag{3.23}$$

For $\hat{Q} = \mathbb{R}^{2d} \times (t,T)$ we call $\underline{u} \colon \hat{Q} \to U$ a feedback control, and if for $u(s) := \underline{u}(y(s),s)$ there holds $u(s) \in U$ for $s \in (t,T)$ we call $\underline{u}^*$ admissible. Let $\underline{u}^*$ be an admissible feedback control with

$$\underline{u}^*(x,s) \in g^*(x,s) \tag{3.24}$$

for all $(x,s) \in \hat{Q}$ 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

$$u^*(s) = \underline{u}^*(y^*(s), s). \tag{3.25}$$

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

Proposition 3.1. *The value function is differentiable in (x,t) if and only if (x,t) is a regular point, i.e. there exists a unique optimal trajectory $y^*(\cdot)$ starting in x at time t with $y^*(t) = x$.*

For a proof we refer to [18, p. 42, Thm. 10.2].

Note that in a regular point the optimal feedback control is given by

$$\underline{u}^*(y^*(s), s) = P_U\left(-\frac{1}{\alpha}\mathcal{B}^T \nabla_x v(y^*(s), s)\right) \qquad \text{(wave equation)}, \tag{3.26}$$

$$\underline{u}^*(y^*(s), s) = P_U\left(-\frac{1}{\alpha} y^*(s)^T \mathcal{B}^T \nabla_x v(y^*(s), s)\right) \qquad \text{(Schrödinger equation)}, \tag{3.27}$$

where P_U denotes the projection on the set of admissible controls.

3.2. **Curse of dimensionality.** Optimal control problems of type (3.18) – (3.19) allow to derive controls in feedback form from the corresponding value function given as the unique viscosity solution of an instationary HJB equation (3.21). 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. To make the problem numerically feasible and to reduce the computational effort two different aspects are crucial. On the one hand reduction of the dimension of the underlying system and on the other hand efficient schemes for solving the HJB equation (3.21). 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 [37], 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. Let here, for simplicity, $Q = [0,1]^d$, $d \in \mathbb{N}$. For a multiindex $l = (l_1, \ldots, l_d) \in \mathbb{N}^d$ we introduce a mesh $Q_{\underline{l}}$ with mesh parameters $h_{\underline{l}} = (h_{l_1}, \ldots, h_{l_d})$ 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

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

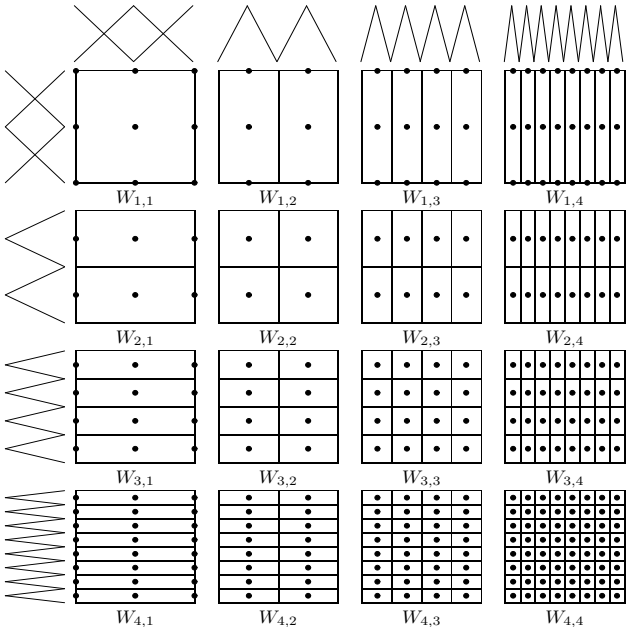


FIGURE 1. Hierarchical spaces for $V_{(4,4)}$

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

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

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

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

with t-th unit vector $\underline{e_t}$. Thus we define

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

cf. Figure 1. With $V_n = V_{(n,\ldots,n)}$ 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

$$\mathcal{B}_{\underline{l}} := \left\{ \underline{j} \in \mathbb{N}^d \;\middle|\; \begin{array}{lll} j_t = 1, \ldots, 2^{l_t} - 1, & j_t \text{ odd}, & t = 1, \ldots, d, \text{ if } l_t > 1, \\ j_t = 0, 1, 2, & & t = 1, \ldots, d, \text{ if } l_t = 1 \end{array} \right\}. \tag{4.6}$$

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

Further we introduce the H^2_{mix} norm and semi-norm given by

$$\|f\|^2_{H^2_{\text{mix}}(Q)} = \sum_{0\leq \underline{k}\leq 2} |\partial^{|\underline{k}|_1}_{x_1^{k_1}\ldots x_d^{k_d}} f|_2^2, \quad \text{and} \quad |f|_{H^2_{\text{mix}}(Q)} = \left\| \partial^{2d}_{x_1^2\ldots x_d^2} f \right\|_2, \tag{4.7}$$

respectively, and we define $H^2_{\text{mix}}(Q) = \left\{ f \in H \mid \|f\|_{H^2_{\text{mix}}} \leq C \text{ for } C > 0 \right\}$ which satisfies the relation $H^4(Q) \subset H^2_{\text{mix}}(Q) \subset H^2(Q)$, see [26]. For $f \in H^2_{\text{mix}}(Q)$ there holds

$$\left\| f_{\underline{l}} \right\|_2 \leq C(d) \cdot 2^{-2|\underline{l}|_1} |f|_{H^2_{\text{mix}}(Q)} \tag{4.8}$$

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

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

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, see estimate (4.8). Following Griebel [24] and Zenger [52], we replace $\|\underline{l}\|_\infty \leq n$ by the rule

$$\|\underline{l}\|_1 \leq n + d - 1. \tag{4.9}$$

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 $\dim \mathcal{O}(2^{nd})$. For functions $f \in H^2_{\text{mix}}(Q)$ there holds the error estimate

$$\|f - f_n^s\|_2 = O(h_n^2 (\log h_n^{-1})^{d-1})) \tag{4.10}$$

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

$$\|f - f_n^s\|_2 = \mathcal{O}(h_n^2). \tag{4.11}$$

This fact leads to a strong reduction of the computational storage consumption in comparison to a full mesh approach. If the required H^s_{mix}-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 [9], where the values of the hierarchical coefficients are employed as error indicators [17, 48].

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 $\varepsilon > 0$,

$$|a_{\underline{l},\underline{j}}| \left\| \phi_{\underline{l},\underline{j}} \right\| > \varepsilon \tag{4.12}$$

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 [9] 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* $\tilde{\underline{l}} = \underline{l} + \underline{e}_t$ *and* $\tilde{\underline{j}} \in \{\underline{j} + j_t\ \underline{e}_t \pm 1\}$ **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 $\tilde{\underline{l}} \le \underline{l}$ and $\operatorname{supp}(\phi_{\tilde{\underline{l}},\tilde{\underline{j}}}) \cap \operatorname{supp}(\phi_{\underline{l},\underline{j}}) \neq \emptyset$;
  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

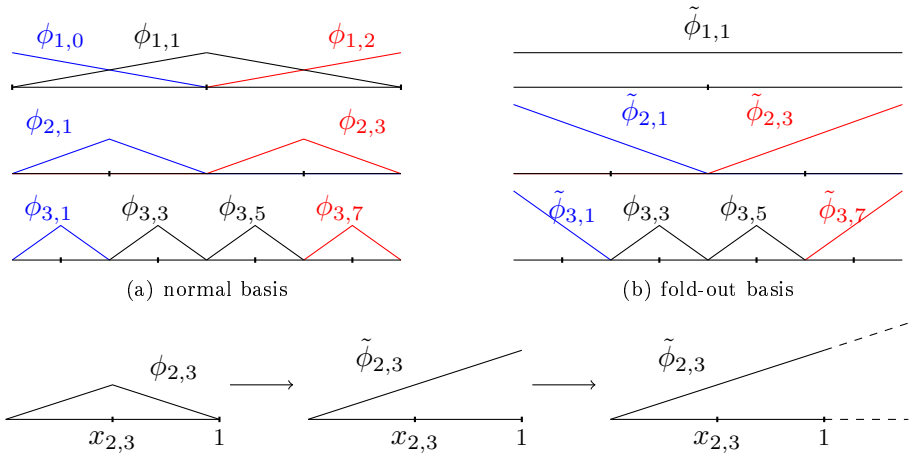
$$|a_{\underline{l},\underline{j}}| \left\| \phi_{\underline{l},\underline{j}} \right\| < \eta, \tag{4.13}$$

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 [9] for more details. A typical choice is $\eta = \varepsilon/10$, which we will use in our experiments in Section 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**
        $\mathcal{I} = \mathcal{I} \backslash (\underline{l},\underline{j})$ ▷ `remove if no children in` $\mathcal{I}$

Note that several norms are possible in (4.12) and (4.13), 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.



(a) normal basis (b) fold-out basis

(c) last basis before the boundary gets folded up and linearly extrapolated across the boundary

FIGURE 2. Normal and fold out 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 $Q_{\mathcal{I}}$, where $\mathcal{I}$ is either the index set of a regular or adaptive sparse grid, the procedure

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

for all $\underline{x} \in Q_{\mathcal{I}}$, time step $\Delta t > 0$, $k = K, \ldots, 0$, and $K = T/\Delta t$. 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., [48]. 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 [9, 48], see Figure 2(a-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 Figure 2(c). 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 $(x, y) \in \partial Q_l$ and $h > 0$ the distance from $\tilde{x}$ to Q_l, 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.

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 [34]. For global minimization algorithms on sparse grids see, e.g., [46]. 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 choose

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

see, e.g., [3, 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 (3.26) and (3.27). 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

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

for the wave equation and

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

for the Schrödinger equation, respectively, with

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

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 Section 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

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

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

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

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 $k = K, ..., 0$

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 $\mathcal{I}(0)$, η, and v_0 ▷ `coarsen` v_0

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

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

  Alg. 2 with $\mathcal{I}(k-1)$, η 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 (3.19) 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. The stepsize h for computing (5.6) is chosen differently from the choice within the semi-Lagrangian scheme, but has to be chosen carefully since numerically we observe a strong sensitivity with respect to the size of h. In particular for small mesh sizes errors in the evaluation are scaled and may cause large errors.

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 (3.18) – (3.19) 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 Section 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". Since we only have an inefficient proof-of-concept sparse grid implementation available, we abstain from giving computation times. Using efficient sparse grid implementations [12, 48] would significantly (an order of magnitude or more) change the needed runtime.

While on regular grids various error estimates for semi-Lagrangian schemes are known, see, e.g., [3, Appendix A] and [4], on sparse grids very little results are available in the literature, see, e.g., [51] where under a certain assumptions an estimate is derived.

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 [8, 47]. We compute a reference solution v_{ref} by an ENO scheme as a variant of a Lax-Friedrichs scheme

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

where $\mathcal{H}_{LF}$ is the numerical Hamiltonian, $D^{\pm}v^k_{\text{ref}}(x_I)$ 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 [37]. For the numerical realization of the scheme we use the software library ROC-HJ (see [7]).

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

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

and set the feedback operator as $\underline{u}(x,t) = -\mathcal{R}^{-1}\mathcal{B}^T P(t)x$ with $\mathcal{R} = \alpha$.

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

$$e_2^v := \|v - v^*\|_{L^2(Q)} \approx \frac{\|v - v_r\|_2}{\sqrt{N_r}}, \tag{6.3}$$

$$e_\infty^v := \|v - v^*\|_{L^\infty(Q)} \approx \|v - v_r\|_\infty. \tag{6.4}$$

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

$$e_2^y := \|y - y^*\|_{L^2(0,T;\mathbb{R}^{2d})}, \quad e_\infty^y := \|y - y^*\|_{L^\infty(0,T;\mathbb{R}^{2d})}, \tag{6.5}$$

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

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

$$\rho_e(i) = \log_2\left(\frac{e_{i-1}}{e_i}\right), \tag{6.7}$$

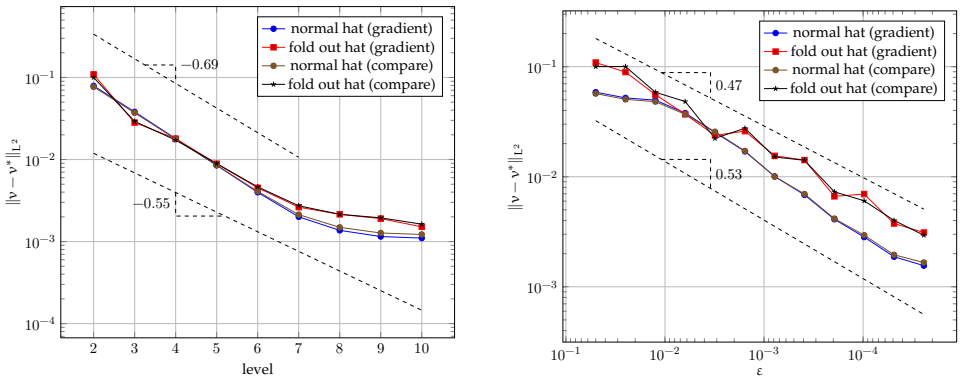
where e_i denotes the error on level i or the error for $\varepsilon = 0.1 \cdot 1/2^i$, respectively, and e is any one of the above errors.

In the following experiments, we use for the feedback law by comparison a discretization of the control space with 40 equidistantly spaced controls in one dimension, while for the feedback law by using the gradient we use the stepsize $\Delta x = \frac{1}{40}$.

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 (3.12) with

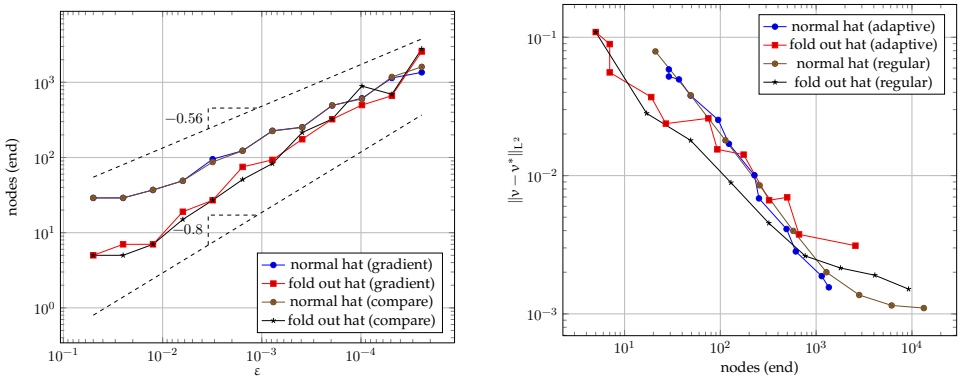
$$\beta_x = 2,\ \beta_u = 0.1, \quad T = 1,\ \Delta t = 0.01, \quad \mathcal{A} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \mathcal{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{6.8}$$

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 (3.21) and compare different variants of our semi-Lagrangian scheme on sparse grids. The reference value function is computed as described in Section 6.1 on the domain $[-1,1]^2$ with $\Delta x = 1/200 \cdot \text{length}(Q)$ and $\Delta t = T/1000$.



(a) error in the value function vs. level

(b) error in the value function vs. ε



(c) nodes vs. ε

(d) error in the value function vs. nodes, feedback law based on gradient approach

Figure 3. 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) (2D).

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 Figure 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 Section 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 Figure 3(a) 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 uses less points, it is the preferred choice in this

	normal			fold out		
level	e_2^v	e_∞^v	nodes (end)	e_2^v	e_∞^v	nodes (end)
2	$7.91{\cdot}10^{-2}$	$1.09{\cdot}10^{-1}$	21	$1.09{\cdot}10^{-1}$	$2.68{\cdot}10^{-1}$	5
3	$3.81{\cdot}10^{-2}$	$5.29{\cdot}10^{-2}$	49	$2.82{\cdot}10^{-2}$	$4.02{\cdot}10^{-2}$	17
4	$1.80{\cdot}10^{-2}$	$2.66{\cdot}10^{-2}$	113	$1.80{\cdot}10^{-2}$	$4.19{\cdot}10^{-2}$	49
5	$8.50{\cdot}10^{-3}$	$1.44{\cdot}10^{-2}$	257	$8.89{\cdot}10^{-3}$	$3.07{\cdot}10^{-2}$	129
6	$3.98{\cdot}10^{-3}$	$8.72{\cdot}10^{-3}$	577	$4.53{\cdot}10^{-3}$	$2.42{\cdot}10^{-2}$	321
7	$2.00{\cdot}10^{-3}$	$5.93{\cdot}10^{-3}$	1,281	$2.62{\cdot}10^{-3}$	$2.04{\cdot}10^{-2}$	769
8	$1.37{\cdot}10^{-3}$	$5.75{\cdot}10^{-3}$	2,817	$2.15{\cdot}10^{-3}$	$2.06{\cdot}10^{-2}$	1,793
9	$1.15{\cdot}10^{-3}$	$3.84{\cdot}10^{-3}$	6,145	$1.90{\cdot}10^{-3}$	$1.89{\cdot}10^{-2}$	4,097
10	$1.10{\cdot}10^{-3}$	$3.45{\cdot}10^{-3}$	13,313	$1.51{\cdot}10^{-3}$	$1.25{\cdot}10^{-2}$	9,217

Table 1. Error in the value function with regular sparse grids using both kinds of basis functions and the gradient approach (2D).

	gradient			compare		
ε	e_2^v	e_∞^v	nodes (end)	e_2^v	e_∞^v	nodes (end)
5.00_{-2}	$5.86{\cdot}10^{-2}$	$7.54{\cdot}10^{-2}$	29	$5.69{\cdot}10^{-2}$	$7.45{\cdot}10^{-2}$	29
2.50_{-2}	$5.20{\cdot}10^{-2}$	$6.44{\cdot}10^{-2}$	29	$5.05{\cdot}10^{-2}$	$6.31{\cdot}10^{-2}$	29
1.25_{-2}	$4.96{\cdot}10^{-2}$	$6.18{\cdot}10^{-2}$	37	$4.81{\cdot}10^{-2}$	$5.95{\cdot}10^{-2}$	37
6.25_{-3}	$3.79{\cdot}10^{-2}$	$5.13{\cdot}10^{-2}$	49	$3.69{\cdot}10^{-2}$	$4.95{\cdot}10^{-2}$	49
3.13_{-3}	$2.53{\cdot}10^{-2}$	$3.73{\cdot}10^{-2}$	95	$2.56{\cdot}10^{-2}$	$3.86{\cdot}10^{-2}$	87
1.56_{-3}	$1.70{\cdot}10^{-2}$	$2.51{\cdot}10^{-2}$	123	$1.71{\cdot}10^{-2}$	$2.53{\cdot}10^{-2}$	123
7.81_{-4}	$1.01{\cdot}10^{-2}$	$1.63{\cdot}10^{-2}$	227	$1.00{\cdot}10^{-2}$	$1.60{\cdot}10^{-2}$	225
3.91_{-4}	$6.84{\cdot}10^{-3}$	$1.03{\cdot}10^{-2}$	253	$6.97{\cdot}10^{-3}$	$1.04{\cdot}10^{-2}$	251
1.95_{-4}	$4.11{\cdot}10^{-3}$	$6.40{\cdot}10^{-3}$	491	$4.16{\cdot}10^{-3}$	$6.46{\cdot}10^{-3}$	495
9.77_{-5}	$2.83{\cdot}10^{-3}$	$4.51{\cdot}10^{-3}$	613	$2.94{\cdot}10^{-3}$	$4.61{\cdot}10^{-3}$	601
4.88_{-5}	$1.88{\cdot}10^{-3}$	$3.45{\cdot}10^{-3}$	1,143	$1.95{\cdot}10^{-3}$	$3.92{\cdot}10^{-3}$	1,179
2.44_{-5}	$1.56{\cdot}10^{-3}$	$5.31{\cdot}10^{-3}$	1,357	$1.66{\cdot}10^{-3}$	$5.60{\cdot}10^{-3}$	1,607

Table 2. Error in the value function for adaptively refined sparse grids using compare and gradient approaches with the normal basis functions (2D).

setting. 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 $Q = [-1,1]^2$. In this case we observe overall similar convergence behaviour to before, but essentially identical errors for the normal and fold out basis functions, which is due to the fact that the treatment of the boundary of the enlarged domain has little effect on the value function in Q.

From Table 2 and Figure 3(b) 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 use less basis functions, see Figure 3(c). When comparing the

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

Table 3. Error in the trajectory and control with regular sparse grids using the gradient approach and fold out basis functions (2D).

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

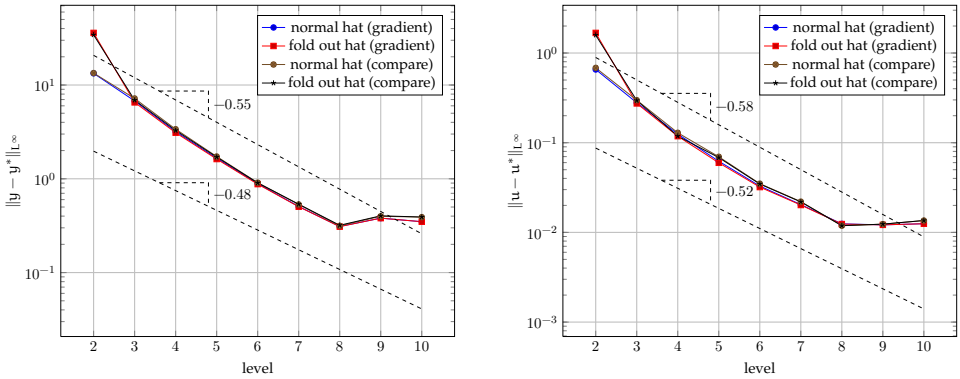
Table 4. Error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions (2D).

error reduction with increasing number of nodes for adaptive versus regular grids, we see in Figure 3(d), 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.

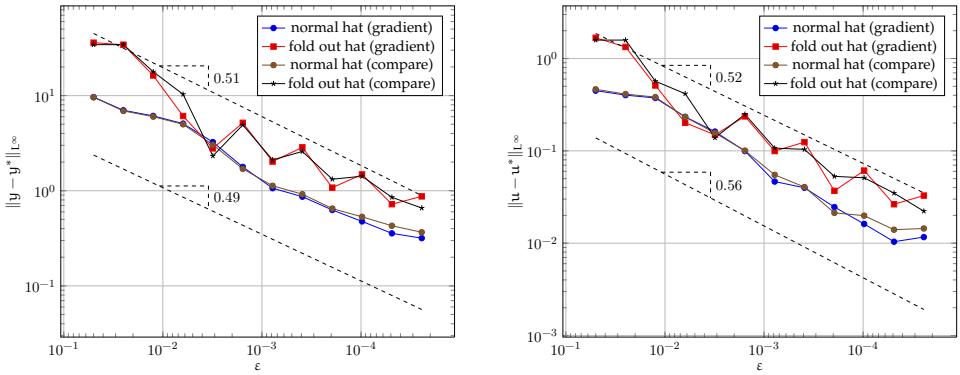
In Table 3 we see the error in the state and control with respect to the L^2- and L^∞-norm and the corresponding rates of convergence on regular sparse grids with the gradient approach and fold out basis functions. The rate of convergence shows a similar behaviour in all three variables with values from 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 Figure 4. When using fold



(a) error in the trajectory vs. level (b) error in the control vs. level

Figure 4. Comparing the scheme on regular sparse grids using normal hat and fold out basis functions (2D).

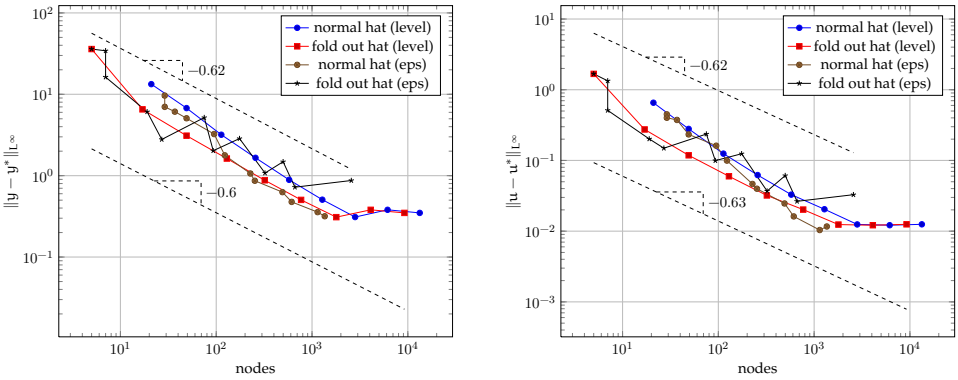


(a) error in the trajectory vs. ε (b) error in the control vs. ε

Figure 5. Comparing the scheme on adaptive sparse grids using normal hat and fold out basis functions (2D).

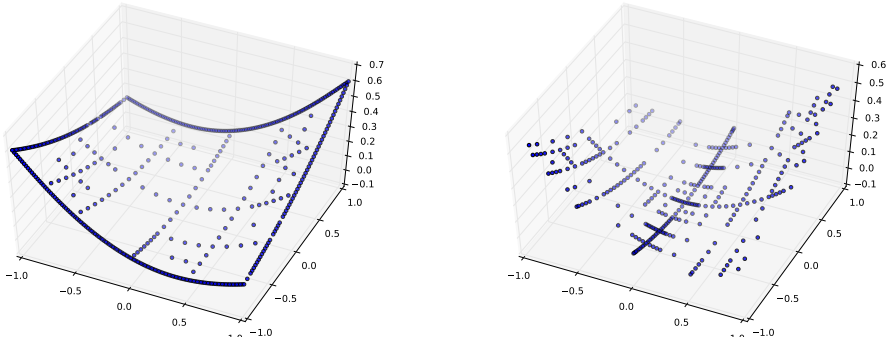
out hat functions in comparison to normal ones we observe a slight reduction of the convergence rate for the feedback law by comparison and the amplitude of the oscillations increases when using the feedback law based on the gradient on adaptive sparse grids, see Figure 5 and Table 4. In Figure 6 the 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 adaptive sparse grids. The corresponding adaptive sparse mesh at initial time with normal and fold out basis functions is shown in Figure 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.



(a) error in the trajectory vs. nodes (b) error in the control vs. nodes

Figure 6. Comparing the scheme using the gradient approach (5.2.2) on adaptive and regular sparse grids using normal hat and fold out basis functions (2D).



(a) normal hat functions (b) fold-out hat functions

Figure 7. Adaptive sparse grid for $\varepsilon = 1.95 \cdot 10^{-4}$ (2D).

6.3. **Control of the semi-discrete wave equation (4D).** In this example we consider the optimal control problem (3.18)–(3.19) for the semi-discrete wave equation (3.12) in dimension four, that means we discretize the state and velocity by two modes using (3.9). We choose the parameters as

$$\beta_x = 2, \quad \beta_u = 0.1, \quad T = 4, \quad \Delta t = 0.01, \quad c = 0.05. \tag{6.9}$$

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 Table 5 and Table 6 the behaviour 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 Figure 8(a) 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,

level	e^y_∞	$\rho_{e^y_\infty}$	e^u_∞	$\rho_{e^u_\infty}$	nodes (end)
2	$6.93{\cdot}10^{-1}$	–	$2.87{\cdot}10^{0}$	–	9
3	$6.27{\cdot}10^{-1}$	0.14	$1.79{\cdot}10^{0}$	0.68	49
4	$4.40{\cdot}10^{-1}$	0.51	$1.36{\cdot}10^{0}$	0.40	209
5	$2.61{\cdot}10^{-1}$	0.75	$5.72{\cdot}10^{-1}$	1.24	769
6	$1.45{\cdot}10^{-1}$	0.85	$2.95{\cdot}10^{-1}$	0.96	2,561
7	$9.00{\cdot}10^{-2}$	0.68	$1.95{\cdot}10^{-1}$	0.59	7,937
8	$4.53{\cdot}10^{-2}$	0.99	$9.47{\cdot}10^{-2}$	1.05	23,297
9	$1.42{\cdot}10^{-2}$	1.68	$3.28{\cdot}10^{-2}$	1.53	65,537
10	$6.01{\cdot}10^{-3}$	1.24	$1.43{\cdot}10^{-2}$	1.19	$1.78 \cdot 10^5$

Table 5. Error in the trajectory and control with regular sparse grids using the gradient approach and fold out basis functions (4D).

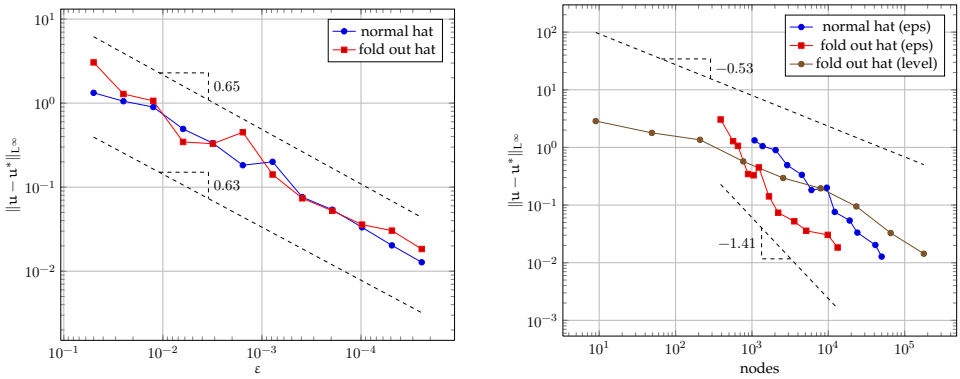
ε	e^y_∞	$\rho_{e^y_\infty}$	e^u_∞	$\rho_{e^u_\infty}$	nodes (end)
5.00_{-2}	$3.67{\cdot}10^{-1}$	–	$3.05{\cdot}10^{0}$	–	389
2.50_{-2}	$2.52{\cdot}10^{-1}$	0.54	$1.28{\cdot}10^{0}$	1.25	567
1.25_{-2}	$1.38{\cdot}10^{-1}$	0.87	$1.07{\cdot}10^{0}$	0.27	655
6.25_{-3}	$1.35{\cdot}10^{-1}$	0.03	$3.46{\cdot}10^{-1}$	1.62	889
3.13_{-3}	$1.39{\cdot}10^{-1}$	−0.04	$3.29{\cdot}10^{-1}$	0.07	1,051
1.56_{-3}	$1.41{\cdot}10^{-1}$	−0.02	$4.50{\cdot}10^{-1}$	−0.45	1,231
7.81_{-4}	$8.73{\cdot}10^{-2}$	0.70	$1.42{\cdot}10^{-1}$	1.67	1,671
3.91_{-4}	$6.33{\cdot}10^{-2}$	0.46	$7.37{\cdot}10^{-2}$	0.94	2,207
1.95_{-4}	$4.53{\cdot}10^{-2}$	0.48	$5.23{\cdot}10^{-2}$	0.49	3,581
9.77_{-5}	$3.09{\cdot}10^{-2}$	0.55	$3.59{\cdot}10^{-2}$	0.54	5,133
4.88_{-5}	$1.94{\cdot}10^{-2}$	0.67	$3.04{\cdot}10^{-2}$	0.24	9,829
2.44_{-5}	$1.27{\cdot}10^{-2}$	0.61	$1.84{\cdot}10^{-2}$	0.73	13,249

Table 6. Error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions (4D).

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 Figure 8(b) 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 Table 5 and Table 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.

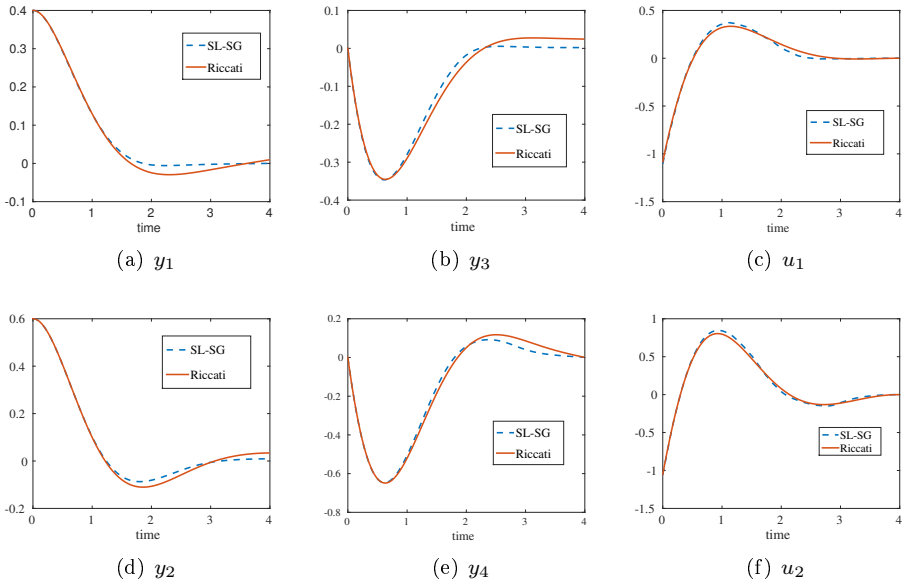
In Figure 9 we confirm that the feedback control we derive from the value function leads to a trajectory which coincide (up to a small error depending on the mesh parameter) with a reference trajectory derived from a Riccati equation.

6.4. **Control of the semi-discrete wave equation (6D).** Next, we consider problem (3.18)–(3.19) for the semi-discrete wave equation (3.12) in six dimensions



(a) error in the control vs. ε (b) error in the control vs. nodes

FIGURE 8. Error in the control comparing the scheme based on sparse grids using normal hat and fold out basis functions, regular and adaptive sparse grids (4D).



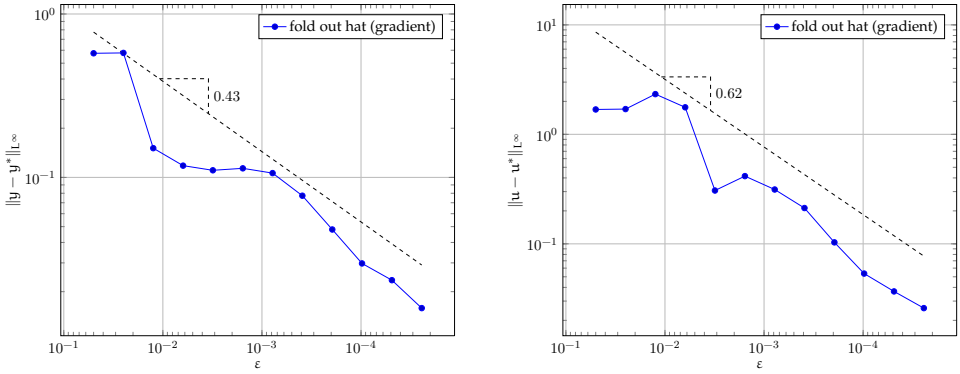
(a) y_1 (b) y_3 (c) u_1

(d) y_2 (e) y_4 (f) u_2

FIGURE 9. Components of the optimal state y and corresponding optimal control u resulting from the SL-SG approach on a sparse grid with threshold $\varepsilon = 2.44 \cdot 10^{-4}$. As a reference solution the corresponding components of a solution generated by a Riccati approach are presented (4D).

ε	e^y_∞	$\rho_{e^y_\infty}$	e^u_∞	$\rho_{e^u_\infty}$	nodes (end)
5.00_{-2}	$5.75{\cdot}10^{-1}$	–	$1.69{\cdot}10^{0}$	–	43
2.50_{-2}	$5.78{\cdot}10^{-1}$	-0.01	$1.70{\cdot}10^{0}$	-0.01	69
1.25_{-2}	$1.51{\cdot}10^{-1}$	1.94	$2.33{\cdot}10^{0}$	-0.45	1,015
6.25_{-3}	$1.18{\cdot}10^{-1}$	0.36	$1.77{\cdot}10^{0}$	0.40	919
3.13_{-3}	$1.11{\cdot}10^{-1}$	0.09	$3.07{\cdot}10^{-1}$	2.52	1,549
1.56_{-3}	$1.14{\cdot}10^{-1}$	-0.04	$4.15{\cdot}10^{-1}$	-0.44	1,779
7.81_{-4}	$1.06{\cdot}10^{-1}$	0.10	$3.13{\cdot}10^{-1}$	0.41	2,053
3.91_{-4}	$7.73{\cdot}10^{-2}$	0.46	$2.12{\cdot}10^{-1}$	0.56	2,851
1.95_{-4}	$4.80{\cdot}10^{-2}$	0.69	$1.03{\cdot}10^{-1}$	1.04	3,657
9.77_{-5}	$2.98{\cdot}10^{-2}$	0.69	$5.35{\cdot}10^{-2}$	0.95	5,415
4.88_{-5}	$2.35{\cdot}10^{-2}$	0.34	$3.68{\cdot}10^{-2}$	0.54	9,691
2.44_{-5}	$1.59{\cdot}10^{-2}$	0.57	$2.59{\cdot}10^{-2}$	0.51	12,507

Table 7. Error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions (6D).



(a) error in the trajectory vs. ε

(b) error in the control vs. ε

Figure 10. The adaptive sparse grids scheme using fold out basis functions (6D).

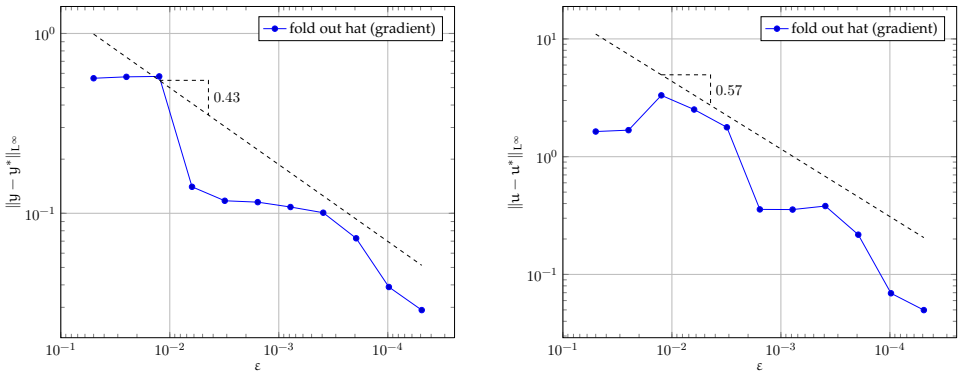
with data as given in (6.9) and initial point $x = (0.4, 0.6, 0.2, 0, 0, 0)^T$. Note that we have to decrease the size of the time step to $\Delta t = 0.0025$ when increasing the entries in the stiffness matrix. We remark that larger entries in the stiffness matrix lead to an increase of the derivative $|H_p(x,p)|$, 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 computational cost low. In Table 7 and Figure 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. The number of mesh points on the finest refinement level is still moderate, however the computational cost is relatively high.

6.5. **Control of the semi-discrete wave equation (8D).** Next, we consider problem (3.18)–(3.19) for the semi-discrete wave equation (3.12) in eight dimension

ε	e^y_∞	$\rho_{e^y_\infty}$	e^u_∞	$\rho_{e^u_\infty}$	nodes (end)
5.00_{-2}	$5.64{\cdot}10^{-1}$	–	$1.63{\cdot}10^{0}$	–	49
2.50_{-2}	$5.74{\cdot}10^{-1}$	-0.03	$1.68{\cdot}10^{0}$	-0.04	63
1.25_{-2}	$5.77{\cdot}10^{-1}$	-0.01	$3.32{\cdot}10^{0}$	-0.98	241
6.25_{-3}	$1.40{\cdot}10^{-1}$	2.04	$2.51{\cdot}10^{0}$	0.40	1,391
3.13_{-3}	$1.17{\cdot}10^{-1}$	0.26	$1.78{\cdot}10^{0}$	0.50	1,455
1.56_{-3}	$1.15{\cdot}10^{-1}$	0.03	$3.57{\cdot}10^{-1}$	2.32	2,169
7.81_{-4}	$1.08{\cdot}10^{-1}$	0.09	$3.56{\cdot}10^{-1}$	0.00	2,759
3.91_{-4}	$1.01{\cdot}10^{-1}$	0.11	$3.81{\cdot}10^{-1}$	-0.10	3,283
1.95_{-4}	$7.26{\cdot}10^{-2}$	0.47	$2.18{\cdot}10^{-1}$	0.81	5,011
9.77_{-5}	$3.89{\cdot}10^{-2}$	0.90	$6.92{\cdot}10^{-2}$	1.65	7,113
4.88_{-5}	$2.90{\cdot}10^{-2}$	0.43	$4.98{\cdot}10^{-2}$	0.47	9,059

Table 8. Error in the trajectory and control with adaptive grids using the gradient approach and fold out basis functions (8D).

with data as given in (6.9) 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 Figure 11 we see the error 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.



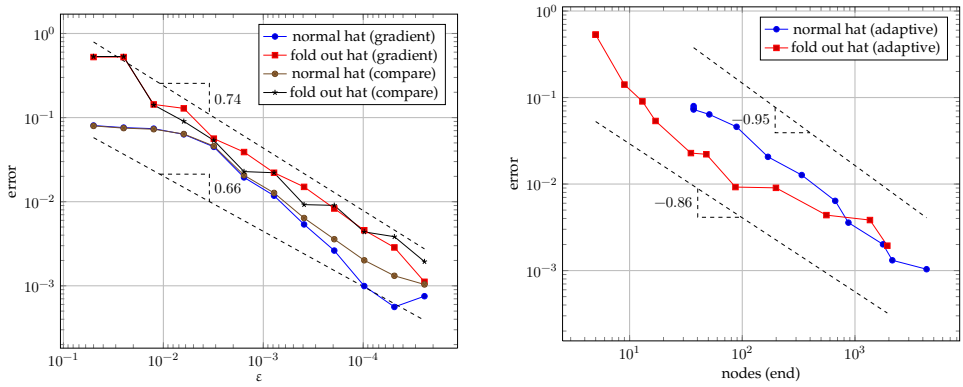
(a) error in the trajectory vs. ε

(b) error in the control vs. ε

Figure 11. The adaptive sparse grids scheme using fold out basis functions (8D).

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. 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.



(a) error in the value function vs. ε (b) error in the value function vs. nodes

FIGURE 12. Bilinear equation (6.10): Error in the value function comparing the scheme based on sparse grids using normal and fold out basis functions, adaptive sparse grid (2D).

Example 1. Let $T = 1, \Delta t = 1/200, \beta_x = 2, \beta_u = 0.1,$

$$\mathcal{A} = \begin{pmatrix} 0 & 0.1 \\ -0.1 & 0 \end{pmatrix}, \quad \mathcal{B} = \begin{pmatrix} 0 & -0.1 \\ 0.1 & 0 \end{pmatrix}, \quad \mathcal{M} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{6.10}$$

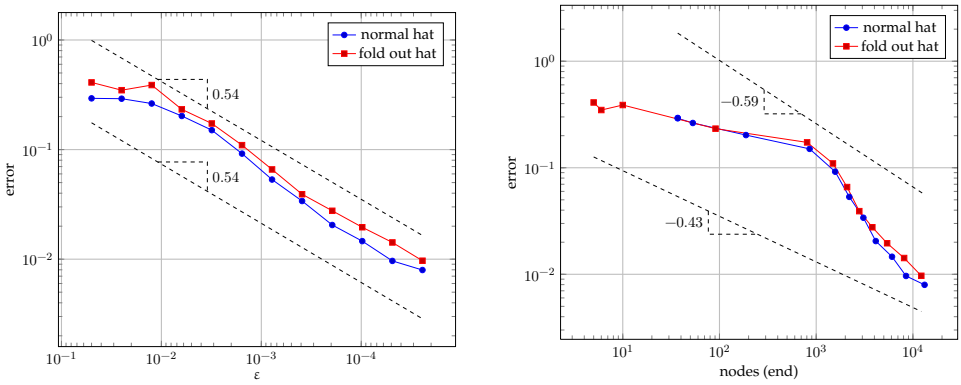
and control bounds inactive, i.e. we choose $U_s = [-10, 10]$ big enough for the comparison approach. For computing a reference value function as described in Section 6.1 we choose Δx and Δt as given in Section 6.2. In Figure 12 we see the error when using comparison and the gradient. Using normal/fold 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

$$\mathcal{A} = \begin{pmatrix} 0 & 0.5 \\ -0.5 & 0 \end{pmatrix}, \quad \mathcal{B} = \begin{pmatrix} 0 & -0.5 \\ 0.5 & 0 \end{pmatrix}, \tag{6.11}$$

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 Figure 13 we see the behaviour of the error when using the comparison approach and normal and fold out basis functions, respectively. In Figure 14 the meshes for $\varepsilon = 1.95 \cdot 10^{-4}$ and $\varepsilon = 7.78 \cdot 10^{-4}$ are shown. We observe a strong refinement of the mesh along the non-differentiability.

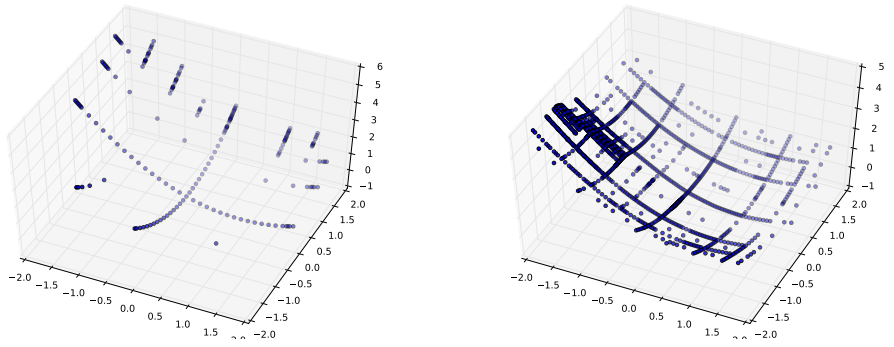
In both example we observe a similar convergence behaviour as for the wave equation. Furthermore, in this two dimensional setting again the different approaches lead to quite the same results.



(a) error in the value function vs. ε

(b) error in the value function vs. nodes

FIGURE 13. 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) Example (6.10) with feedback (5.3), fold-out basis functions and $\varepsilon = 1.95 \cdot 10^{-4}$

(b) Example (6.11) with feedback (5.3), fold-out basis functions and $\varepsilon = 7.78 \cdot 10^{-4}$

FIGURE 14. Bilinear equation: Sparse grid (2D).

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 [51]). 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 [48] 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] A. Alla and M. Falcone, *An adaptive POD approximation method for the control of advection-diffusion equations*, International Series of Numerical Mathematics **164** (2013), 1–17.

[2] A. Alla, M. Falcone, and D. Kalise, *HJB-POD based feedback design approach for the wave equation*, (2015), preprint, RICAM Report 15-09, `https://www.ricam.oeaw.ac.at/publications/reports/15/rep15-09.pdf`.

[3] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhäuser, Boston, 2008.

[4] G. Barles and P.E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymptotic Analysis **4** (1991), no. 3, 271–283.

[5] L. Baudouin and J. Salomon, *Constructive solution of a bilinear optimal control problem for a Schrödinger equation*, Systems & Control Letters **57** (2008), no. 6, 453 – 464.

[6] K. Beauchard, J.M. Coron, and H. Teismann, *Minimal time for the bilinear control of Schrödinger equations*, Systems Control Lett. **71** (2014), 1–6.

[7] O. Bokanowski, A. Desilles, and H. Zidani, *ROC-HJ-Solver. A C++ library for solving HJ equations*, (2013).

[8] O. Bokanowski, N. Forcadel, and H. Zidani, *Reachability and minimal times for state constrained nonlinear problems without any controllability assumption*, SIAM Journal on Control and Optimization **48** (2010), no. 7, 4292–4316.

[9] O. Bokanowski, J. Garcke, M. Griebel, and I. Klompmaker, *An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations*, Journal of Scientific Computing **55** (2013), no. 3, 575–605.

[10] A. Borzi, J. Salomon, and S. Volkwein, *Formulation and numerical solution of finite-level quantum optimal control problems*, Journal of Computational and Applied Mathematics **216** (2008), no. 1, 170 – 197.

[11] H.-J. Bungartz and M. Griebel, *Sparse grids*, Acta Numerica **13** (2004), 1–123.

[12] G. Buse, *Exploiting many-core architectures for dimensionally adaptive sparse grids*, Dissertation, Institut für Informatik, Technische Universität München, München, May 2015.

[13] E. Carlini, M. Falcone, and R. Ferretti, *An efficient algorithm for Hamilton-Jacobi equations in high dimension*, Computing and Visualization in Science **7** (2004), no. 1, 15–29.

[14] A. Chkifa, A. Cohen, and C. Schwab, *High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs*, Foundations of Computational Mathematics **14** (2014), no. 4, 601–633.

[15] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, Applied mathematics, Society for Industrial and Applied Mathematics, 2014.

[16] R. Ferretti, *Internal approximation schemes for optimal control problems in Hilbert spaces*, Journal of Mathematical Systems, Estimation, and Control **7** (1997), no. 1, 1–25.

[17] C. Feuersänger, *Sparse Grid Methods for Higher Dimensional Approximation*, Dissertation, Institut für Numerische Simulation, Universität Bonn, September 2010.

[18] W.H. Fleming and H.M. Soner, *Controlled markov processes and viscosity solutions*, Stochastic Modelling and Applied Probability, Springer New York, 2006.

[19] G. Friesecke, F. Henneke, and K. Kunisch, *Sparse control of quantum systems*, Preprint No. IGDK-2015-21, `http://www.ma.tum.de/foswiki/pub/IGDK1754/Preprints/FrieseckeHennekeKunisch_2015.pdf`.

[20] J. Garcke, *Sparse grids in a nutshell*, Sparse grids and applications (J. Garcke and M. Griebel, eds.), Lecture Notes in Computational Science and Engineering, vol. 88, Springer, 2012, pp. 57–80.

[21] J. Garcke and M. Griebel, *On the computation of the eigenproblems of hydrogen and helium in strong magnetic and electric fields with the sparse grid combination technique*, Journal of Computational Physics **165** (2000), no. 2, 694–716.

[22] M. Gerdts, G. Greif, and H. J. Pesch, *Numerical optimal control of the wave equation: Optimal boundary control of a string to rest in finite time*, Math. Comput. Simulation **79** (2008), no. 4, 1020–1032.

[23] S. Gombao, *Approximation of optimal controls for semilinear parabolic PDE by solving Hamilton-Jacobi-Bellman equations.*, Proc. of the 15th International Symposium on the Mathematical Theory of Networks and Systems, of Notre Dame, South Bend, Indiana, USA, 2002.

[24] M. Griebel, *A parallelizable and vectorizable multi-level algorithm on sparse grids*, Parallel Algorithms for Partial Differential Equations, Notes on Numerical Fluid Mechanics (W. Hackbusch, ed.), vol. 31, Vieweg Verlag, Braunschweig, 1991, pp. 94–100.

[25] M. Griebel and J. Hamaekers, *Sparse grids for the Schrödinger equation*, Mathematical Modelling and Numerical Analysis **41** (2007), no. 2, 215–247.

[26] M. Griebel and S. Knapek, *Optimized general sparse grid approximation spaces for operator equations*, Mathematics of Computations **78** (2009), no. 268, 2223–2257.

[27] M. Gugat and G. Leugering, *L^∞-norm minimal control of the wave equation: on the weakness of the bang-bang principle*, ESAIM: Control, Optimisation and Calculus of Variations **14** (2008), no. 2, 254–283.

[28] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control.*, Dimension Reduction of Large-Scale Systems, Lect. Notes Comput. Sci. Eng., vol. 45, Springer, Berlin, 2005, pp. 261–306.

[29] M.B. Horowitz, A. Damle, and J.W. Burdick, *Linear hamilton jacobi bellman equations in high dimensions*, Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, Dec 2014, pp. 5880–5887.

[30] C. Hu and C. Shu, *A discontinuous Galerkin finite element method for Hamilton–Jacobi equations*, SIAM Journal on Scientific Computing **21** (1999), no. 2, 666–690.

[31] H. Ishii, *Uniqueness of unbounded viscosity solution of Hamilton-Jacobi equations*, Indiana University Mathematics Journal **33** (1984), no. 5, 721–748.

[32] K. Ito and K. Kunisch, *Optimal bilinear control of an abstract Schrödinger equation*, SIAM Journal on Control and Optimization **46** (2007), no. 1, 274–287.

[33] D. Kalise and A. Kröner, *Optimal feedback control of advection-diffusion-reaction system using balanced truncation*, Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems (MTNS), 2013, pp. 1196–1202.

[34] D. Kalise, A. Kröner, and K. Kunisch, *Local minimization in dynamic programming equations*, (2014), preprint, `https://www.ricam.oeaw.ac.at/publications/reports/15/rep15-04.pdf`.

[35] A. Kröner and K. Kunisch, *A minimum effort optimal control problem for the wave equation*, Comput. Optim. Appl. **57** (2014), no. 1, 241–270.

[36] A. Kröner, K. Kunisch, and B. Vexler, *Semi-smooth Newton methods for optimal control of the wave equation with control constraints*, SIAM Journal on Control and Optimization **49** (2011), no. 2, 830–858.

[37] A. Kröner, K. Kunisch, and H. Zidani, *Optimal feedback control of the undamped wave equation by solving a HJB equation*, ESAIM: Control, Optimisation and Calculus of Variations **21** (2015), no. 2, 442 – 464.

[38] K. Kunisch, P. Trautmann, and B. Vexler, *Optimal control of the undamped linear wave equation with measure value controls*, preprint, `http://www.ma.tum.de/foswiki/pub/IGDK1754/Preprints/KunischTrautmannVexler_2014.pdf`, 2014.

[39] K. Kunisch, S. Volkwein, and L. Xie, *HJB-POD based feedback design for the optimal control of evolution problems*, SIAM Journal on Applied Dynamical Systems **4** (2004), 701–722.

[40] K. Kunisch and D. Wachsmuth, *On time optimal control of the wave equation, its regularization and optimality system*, ESAIM: Control, Optimisation and Calculus of Variations **19** (2013), 317–336.

[41] K. Kunisch and L. Xie, *POD-based feedback control of the Burgers equation by solving the evolutionary HJB equation*, Computers and Mathematics with Applications **49** (2005), 1113–1126.

[42] J.-L. Lions and E. Magenes, *Non-homogeneous boundary value problems and applications*, vol. I, Springer, Berlin, 1972.

[43] J.L. Lions, *Optimal control of systems governed by partial differential equations*, Grundlehren der mathematischen Wissenschaften, Springer-Verlag, 1971.

[44] M. Mirrahimi, P. Rouchon, and G. Turinici, *Lyapunov control of bilinear Schrödinger equations*, Automatica **41** (2005), no. 11, 1987–1994.

[45] M. Mirrahimi and R. Van Handel, *Stabilizing feedback controls for quantum systems*, SIAM Journal on Control and Optimization **46** (2007), no. 2, 445–467.

[46] E. Novak and K. Ritter, *Global optimization using hyperbolic cross points*, State of the Art in Global Optimization: Computational Methods and Applications (C. A. Floudas, ed.), Nonconvex Optim. Appl. 7, Kluwer Academic Publishers, Dordrecht, 1996, pp. 19–33.

[47] S. Osher and C.W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM Journal on Numerical Analysis **28** (1991), no. 4, 907–922.

[48] D. Pflüger, *Spatially adaptive sparse grids for high-dimensional problems*, Dissertation, Institut für Informatik, Technische Universität München, München, February 2010.

[49] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Doklady Akademii nauk SSSR **148** (1963), 1042–1043, Russian, Engl. Transl.: Soviet Math. Dokl. 4:240–243, 1963.

[50] R. Springer, *Lösung von Hamilton-Jacobi-Bellman-Gleichungen auf dünnen Gittern*, 2013, Diplomarbeit, University Chemnitz.

[51] X. Warin, *Adaptive sparse grids for time dependent Hamilton-Jacobi-Bellman equations in stochastic control*, (2014), arXiv:1408.4267.

[52] C. Zenger, *Sparse grids*, Parallel Algorithms for Partial Differential Equations (W. Hackbusch, ed.), Notes on Numerical Fluid Mechanics, vol. 31, Vieweg, 1991, pp. 241–251.

Institut for Numerical Simulation, University Bonn, Germany and Fraunhofer SCAI, Sankt Augustin, Germany

E-mail address: `garcke@ins.uni-bonn.de`

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

E-mail address: `axel.kroener@inria.fr`